A Fast and Scalable System to Visualize Contour Gradient from Spatio-temporal Data

A thesis submitted to the
College of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Zonayed Ahmed

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Disclaimer

Reference in this thesis to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building, 110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

Abstract

Changes in geological processes that span over the years may often go unnoticed due to their inherent noise and variability. Natural phenomena such as riverbank erosion, and climate change in general, is invisible to humans unless appropriate measures are taken to analyze the underlying data. Visualization helps geological sciences to generate scientific insights into such long-term geological events. Commonly used approaches such as side-by-side contour plots and spaghetti plots do not provide a clear idea about the historical spatial trends.

To overcome this challenge, we propose an image-gradient based approach called ContourDiff. ContourDiff overlays gradient vector over contour plots to analyze the trends of change across spatial regions and temporal domain. Our approach first aggregates for each location, its value differences from the neighboring points over the temporal domain, and then creates a vector field representing the prominent changes. Finally, it overlays the vectors (differential trends) along the contour paths, revealing the differential trends that the contour lines (isolines) experienced over time.

We designed an interface, where users can interact with the generated visualization to reveal changes and trends in geospatial data. We evaluated our system using real-life datasets, consisting of millions of data points, where the visualizations were generated in less than a minute in a single-threaded execution. We show the potential of the system in detecting subtle changes from almost identical images, describe implementation challenges, speed-up techniques, and scope for improvements. Our experimental results reveal that ContourDiff can reliably visualize the differential trends, and provide a new way to explore the change pattern in spatiotemporal data. The expert evaluation of our system using real-life WRF (Weather Research and Forecasting) model output reveals the potential of our technique to generate useful insights on the spatiotemporal trends of geospatial variables.

Acknowledgements

First of all, I would like to acknowledge my indebtedness and render my warmest thanks to my respected supervisors Dr. Chanchal K. Roy and Dr. Kevin A. Schneider for their constant guidance, advice, motivation and extraordinary patience during my thesis works. They have been extremely supportive and helpful to me throughout my program. I express my heartiest gratitude to Dr. Debajyoti Mondal for his valuable time, constant guidance and encouragement throughout all stages of my thesis. In addition to their expert guidance in conducting my research works, I got all of them beside me in all my difficult stages of my life during the entire time span.

I am thankful to Dr. Sara Sadri and Dr. Louise Arnal, for their valuable feedback on the developed system.

Special thanks to all of the members of the Software Research Lab for their extreme support and important feedback on my work. In particular, I would like to thank Dr. Christopher Dutchyn, Dr. Mohammad Masudur Rahman, Saikat Mandal, C M Khaled Saifullah, Shamima Yeasmin, Dr. Judith Islam, Md Nadim, Muhammad Mainul Hossain, Kawser Wazed Nafi, Amit Kumar Mondal, Tonny Kar, Golam Mostaeen, Rayhan Ferdous, Debasish Chakroborti, Hamid Khodabandehloo, Dr. Farouq Al. Omari, Avijit Bhattacharjee, Naz Zarreen Oishie, Abdul Awal, Md. Shamimur Rahman, Daniel Abediny, and Sristy Sumana Nath.

I am grateful to the Department of Computer Science of University of Saskatchewan for their generous financial support through scholarships, awards, and bursaries that helped me to concentrate more deeply on my thesis work. I would like to render my warmest thanks to all the staffs of the Department for their constant support. In particular, I would like to thank Gwen Lancaster, Sophie Findlay, Shakiba Jalal and Heather Webb.

I would especially like to thank all my friends and seniors who have been extremely helpful and encouraging throughout the period of my thesis work. In particular, I would like to thank Md. Sami Uddin, Golam Mostaeen, Michael Beyene, Gazi Md. Hasnat Zahan, Minhajul Arifin Badhon, Jenia Afrin Jeba, Nazifa Azam Khan, and Md. Aminul Islam.

I am extremely grateful to my family—my father Md. Nurul Haque, my mother Rehena Parveen, and my sister Nahida Sultana —for their inspiration and love at every stage of my life. Without their endless sacrifice, I would not have come this far.

Finally, I would like to acknowledge the contribution of my wife Maliha Mahbub — her presence, support and encouragement, which has always helped me move forward in my life.

I dedicate this to my family; my father Md. Nurul Haque, my mother Rehena Parveen, my sister Nahida Sultana and my wife Maliha Mahbub.

Contents

List of Tables

List of Figures

1 Introduction

This chapter provides an overview of the thesis. First, we set up the context behind our work in section 1.1. Then, we state the research problems and our contributions on those problems in sections 1.2 and 1.3, respectively. Finally, we outline the remainder chapters of the thesis in 1.4.

1.1 Research Context

The increasing availability of geosensors and remote sensing technologies have created an abundance of geospatial data [37]. Most of such data are collected periodically over time. The generation of sheer amount of time-series data has influenced researchers attempt information driven analysis and discoveries. But, there are a lot of challenges associated with geopspatial data handling. The increasing volume is a critical issue and pose a serious challenge in storing, managing, processing, analysing, visualising and verifying the quality of data [85]. Old habits of data representation such as forms and tables becomes unreadable for large data, and interactive and exploratory visualization environments help in making sense of what the data actually contains [34, 47, 66].

Geospatial analysis is not only limited to understanding a particular spatial location. The analysis often involves a comparison between pairs of spatial locations. For example, while examining a landscape or riverbank erosion, it is important to know the region that is likely to experience a higher rate of erosion than others. Landscape erosion is often considered a slow process, e.g., landscapes of Iowa, formed over the last 10,000 to 12,000 years through soil erosion are considered young compared to the Appalachian mountain region, formed over a million of years [106]. Weathering is another example that causes the physical or chemical breakdown of the minerals in rock. Because of slow weathering processes, soil in Antarctica preserves million years old geological characteristics [27]. Similarly, exploration of parameters such as soil moisture and snow accumulation may reveal hidden correlation among other climatic parameters. Comparative analysis across the spatial domain is thus important for geographical exploration. There are numerous such natural phenomena, e.g., monitoring continent and ocean bed relocation, volcanic activities and earthquakes, and impact of climate change on weather parameters, where geographers and hydrologists are interested in finding historical changes across the spatial domain in light of the present-day contour lines (i.e., geospatial lines with the same value or isolines).

This motivated us to explore spatio-temporal data. We portray *spatio-temporal data* as a dataset that has both spatial and temporal properties. This dataset is collected across time and space, and so, a spatio-

temporal event can be defined as a phenomenon that occurs at a certain timestamp t, as well as at a location l. While the spatial dimension is represented as spatial grid structure (x, y), the temporal dimension is represented by a series of timestamps $t_1, t_2, t_3, ..., t_n$ with equal time difference. A spatiotemporal visualization approach can provide a global view of activities or progress, from which evolutions and overall tendencies can be detected [162]. Hence they can be used to infer long-term effects of an incident, and as a model for future assessment and decision making processes. Analyzing spatio-temporal data in different fields such as geospatial data partitioning [127], stock exchange [104], simulation data [42], news articles [60], photo collections [49], and project plans [11] is ubiquitous to gain insight.

Understanding characteristics of geospatial data with respect to a period of time is useful for a number of reasons. Firstly, mining temporal data through visual analytics can lead to the discovery of interesting and meaningful patterns over the years. Secondly, exploration of spatio-temporal parameters such as soil moisture and snow accumulation may reveal hidden correlation among other climatic parameters. Thirdly, geospatial analysis is not only limited to understanding a particular spatial location. The analysis often involves a comparison between pairs of spatial locations. Comparative analysis among different spatial locations can tell us the similarities and dissimilarities among them. For example, while examining a landscape or riverbank erosion, it is important to know the region that is likely to experience a higher rate of erosion than the others. Comparative analysis across the spatial domain is thus important for geographical exploration.

Visual analytics is a powerful tool to understand geospatial data. Common geospatial data visualization approaches include techniques from a variety of sources such as side-by-side contour plots for different timestamps, animation, map iteration, statistical plots and charts, time-cube, exploratory data analysis, cartography, and image processing [161]. A combination of these approaches enable us to connect particular phenomena to a spatial reference system [97]. Moreover, understanding temporal changes in geospatial data provides valuable information about geological and hydrological processes such as landscape changes [26, 87], modelling slow-earthquake [70], regulating river-bank erosion [157], understanding hydrology-driven events based on soil moisture [53], etc. Thus approaches to dissect and gain key insight from large, geospatial data are warranted.

1.2 Motivation and Research Questions

1.2.1 Analyzing Temporal Changes in Spatial Data

Contour plots have been widely used to visualize geospatial information on two-dimensional maps such as elevation visualization on topographic maps [94, 96, 137], graphical displays of thermodynamics [93], travelling distance plots on isochrone maps [109] etc. A contour plot consists of conotur lines and contour intervals. A contour line or isoline represents a fixed threshold value and connects similar-valued data points over a map. A contour interval corresponds to a range of values within the bounds indicated by two successive threshold values. A common approach to visually examine such geo-science processes is to use side-by-side contour

plots for different timestamps. There are two limitations to this approach. With the growing number of timestamps, it is very difficult for users to concentrate on the uniqueness and it demands a high cognitive load [132]. Moreover, they do not provide a good comprehensive picture of the detailed temporal changes. For example, snow accumulation of a particular area over a single timestamp can be adequately explained to users through contour plot. But to find the overall pattern of change over the years in that area, it may not be enough to look at the individual timestamps. Because a similar change pattern may not occur over the whole area and the change in a spatial location may depend on the changes in other neighbouring locations. Although it is possible to display the information with traditional animation or small multiples [20], the users may fail to notice local changes due to inattentional blindness [98] or change blindness [108]. Hence we examine the following research questions:

Research Question 1*: How can we visualize spatio-temporal changes in a single and static view? How can this view be analyzed for different change patterns over different time periods?*

1.2.2 Fast and Scalable Computation

For interactive exploration of large data, it is essential to process data and extract necessary information within a few seconds or even in milliseconds. Processing such a huge dataset of tuple $(latitude, longitude, time)$ can become inherently complex. Early perceptually approximate techniques utilizes some form of data reduction, for example, sampling [41], filtering [128] etc. that causes missing of certain aspects of data including outliers. The visual exploration of large datasets, however, adds another layer of complexity to the visualization problem [102]. Now, one needs to query the dataset based on a set of user inputs, and provide a visual response as quickly as possible, in order not to impact the outcome of the visual exploration [102]. Data-cube based techniques such as Nanocubes [86] and Hashedcubes [111] pre-computes and stores all aggregates of datasets to speed up queries on spatio-temporal data, while imMens [90] leverages GPUs to achieve fast queries over datasets at large scale. But these techniques are designed mostly to analyze discrete spatio-temporal events, rather than gridded data. Hence, we examine the following research questions:

Research Question 2*: How do we initially extract patterns over the large scale data as fast as possible? How do we provide users maximum query throughput with minimum time latency?*

1.2.3 Interactive Exploration

The interface design process is essential to developing interaction tools and techniques for using geospatial visualization tools effectively. One facet of interaction and navigation with representations of geography (shared with architectural representation and distinguishing both from most other forms of information representation) is that the phenomenon being represented is also the one being used for navigation. Therefore, both the interface and the representational structure of (hyper)media are central to the development of more effective and efficient means of transferring knowledge [29]. From the perspective of geographical visualization, interaction design for navigation implies that the user is able to both navigate the visualization itself and

navigate the synthetic geography that it represents. Moreover, this interactive exploration needs to be real-time so that the changes made to the designed visualization can be represented with minimum time delay. The following research question have been introduced through this motivation:

Research Question 3*: How to effectively and efficiently visualize curated information from raw data with a well-designed and interactive user interface?*

1.3 Contribution

Focusing on the aforementioned research problems in terms of spatio-temporal visualization, our study makes three major contributions. Here in this section we briefly present our contribution of the study.

- **Contribution #1** We examine ways to visualize a differential trend in spatiotemporal data. A *differential trend* at a spatial point P is the aggregated value differences of P from its local neighbourhood. Throughout the thesis, we consider two types of trends: increasing and decreasing. The *increasing (Resp., decreasing)* trends are obtained by considering neighbors that have a higher (resp., lower) value than that of P.

  Figure 1.1 illustrates the elevation of three data points p, q, r at different timestamps $t_1, \ldots, t_k$. At a timestamp t_k, the increasing and decreasing trends at point q are shown using the blue and black arrows, respectively. The direction of the black arrow is leftward, which indicates that historically the left neighbour p has had a lower elevation than q. The length of the arrow corresponds to the average value difference over all timestamps. Since the length of the black arrow is larger than that of blue arrow, we know that the average slope between p and q is sharper than the average slope between q and r.

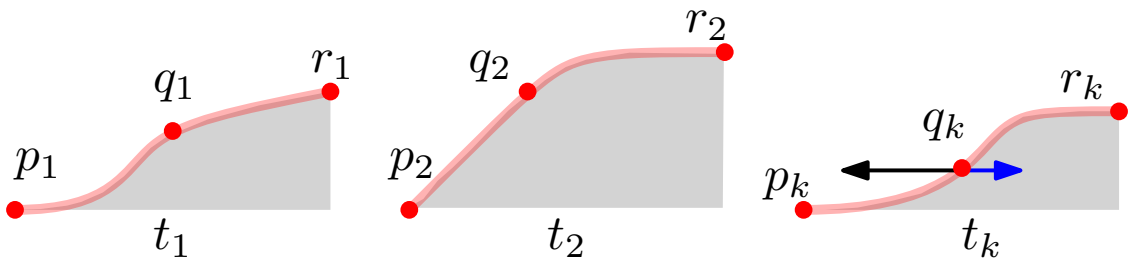

Figure 1.1: Illustration for differential (increasing and decreasing) trends.

  Our idea is simple yet powerful to summarize temporal change information on a single visualization. This can also be seen as a generalization of image gradient [51], which is a fundamental image processing technique to extract information based on the color or intensity changes in a single image. To the best of our knowledge, this is the first attempt to adapt the image gradient approach to visualize changes in spatiotemporal data.

We propose a novel visualization technique, ContourDiff, using this idea to reveal such differential trends. To create the visualization, we first compute a vector field over the spatial domain and overlay that along the simplified contour lines of the region of interest. We do this by creating vector visualization on top of simplified contour map of an area. However, unlike the uniformly spread vector or glyph overlay (e.g., in wind speed or ocean current visualization [158, 61]), we plot the directions along the contour lines, providing an enriched understanding with a much finer resolution and keeping the visualization free of visual clutter. In particular, we first generate the contour map and sampled it using a regular grid such that the users can explore the visualization in various resolutions or zoom levels. We then compute the vector map that represents the differential behavior of every spatial location in relation to their k-hop neighbors, where a k-hop is defined as k rows and k columns surrounding the location. For a single geographic location u, we compute its associated vector directions based on the position of its neighbors in a 5×5 grid. We next aggregate the value differences (between u and its neighbors) over the temporal dimension, and finally, visualize the vectors along the contour lines.

- **Contribution #2** Various cuboid and hierarchical data structures have been analyzed to find an optimized solution ensuring a fast and scalable approach for large scale spatio-temporal data computation. We do not use cuboid structures for their obvious limitations of being lengthy procedures. Moreover, cuboid data structures are mainly used for computation of multi-dimensional spatio-temporal data. Rather than accessing more than 2 dimensions at once, we process dataset of each timestamp sequentially. Thus, computationally, we primarily work on one 2-dimensional (x, y) based data structure at one time. Moreover, we employ a hierarchical spatial data structure, quadtree [123] for recursive decomposition of our 2-dimensional space so that users can store and access data over contour lines quickly. We use PR (Point-Region) quadtree that decomposes the region into four equal quadrants, subquadrants, and so on, until no leaf node contains more than a single point. We use PR quadtree because the cells store a list of points that exist within the cell of a leaf. So, we can easily access the points in a particular region very quickly. To locate all points within radius r of query point Q, we begin at the root. If the root is an empty leaf node, then no data points are found. If the root is a leaf containing a data record, then the location of the data point is examined to determine if it falls within the circle. If the root is an internal node, then the process is performed recursively, but only on those subtrees containing some part of the search circle [7].

- **Contribution #3** We design a Graphical User Interface (GUI) that provides users capability to analyze and understand our ContourDiff visualization. To enable user interactions, we provide a control panel to check data distribution and choose contour thresholds, edit vector mapping and map content properties. We also provide an interactive panel that deploys zooming and panning mechanisms, interchanges different states of the map on user command, and saves the image. The overview of the whole system is portrayed in Figure 1.2.

- **Contribution #4** To examine the usefulness of the change patterns for scientific analysis, we evaluate ContourDiff generated visualizations with two experts. Both experts have extensive knowledge on hydrology modelling and forecasting, and they provide a detailed explanation of various changes in real life meteorological data. Since the experts have been involved in both intermediate and completed versions of the visualization framework, we have discussed the workflow of ContourDiff and how the images are generated. Their prior knowledge on spatiotemporal dataset and various types of visualization from them have provided us with a clear picture on the strengths of our system. Furthermore, we have received a few suggestions that could easily be added as new features to improve ContourDiff.

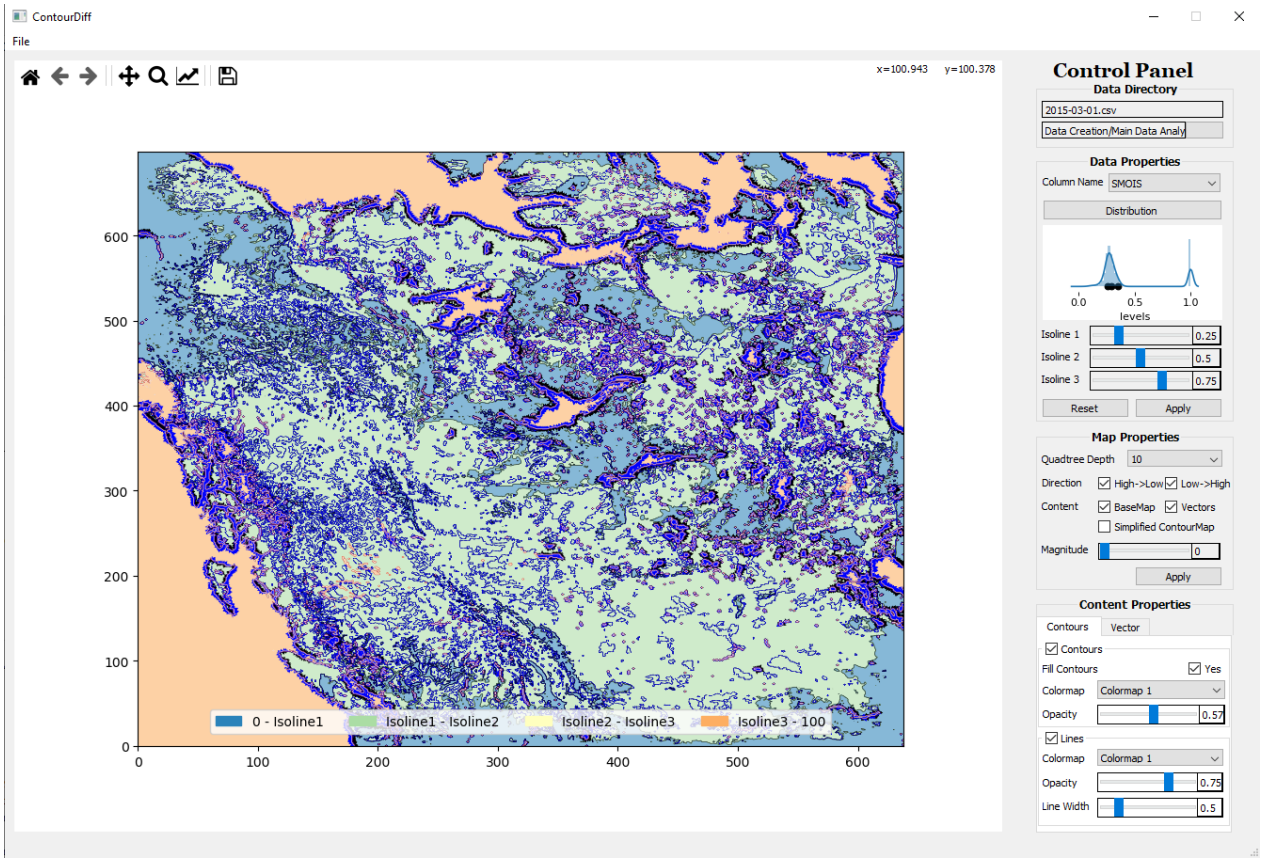

Figure 1.2: Overall system design of ContourDiff and a created visualization.

1.4 Thesis Organization

Our thesis is divided into several chapters to describe and evaluate the process of generating our ContourDiff visualization. Here we discuss the organization of our thesis briefly.

Chapter 2 provides an overview of the related research works in spatio-temporal data. Our thesis is dedicated to explore visualization strategies to extracting patterns from these datasets and portray them on top of contours. So, we discuss the characteristics of spatio-temporal dataset and the changes we can

visualize. We also provide an in-detail exploration of a typical spatio-temporal visualization structure with a workflow, data type exploration, data representation and task analysis. To discuss spatio-temporal data visualization in the literature, we classify them into a Summary-based visualization and Time-varying data visualization. Since, time-varying dataset in a large domain of exploration, we reclassify them as static vs dynamic representations with their strengths and weaknesses. Since we are interested in revealing how data of interest evolve over time, we also analyze temporal trend visualization by view compositions, time-series plots, and pathline demonstrations. We choose pathline demonstration since that provides us the capability to capture and visualize the patterns by vectors. Before exploring that further, the pros and cons of visualization via single integrated view opposed to multiple linked views are introduced. Finally we explore vector mapping and glyph mapping in spatio-temporal data visualization and explain how our visualization context is unique represent with vector mapping.

Chapter 3 discusses different datasets that we use to validate our visualization strategy and find the patterns within them. We describe mainly two types of dataset to explore the patterns in a dataset - real life meteorological dataset and real life image dataset. Since our approach can work with any spatially continuous dataset, we process both numerical and image dataset to work with our system. We also explore the objective of generating a synthetic dataset and how the dataset is designed. Finally we briefly analyze the differences among these data types.

Chapter 4 provides an in-depth analysis for the initial visualization strategies. Since our design is dependant on creating contours from the latest timestamp data, we provide a description of how the contour plot is designed by marching square algorithm and how blending of colormaps or side-by-side visualization can not provide the intended comparative analogy. We also discuss a primitive direct vector overlay technique and its advantages and limitations. Finally we discuss the modifications necessary based on our preliminary observations from this technique.

Chapter 5 discusses how ContourDiff integrates different ideas to create the final visualization. We first describe the techniques to preprocess and filter our different types of dataset. Then we describe the steps to create a contourmap from the latest timestamp. We also discuss how the feature or patterns has been extracted based on the relationship of a data point and its neighbors. Then we discuss how quadtree data structure helps us store and filter our created patterns by user-defined queries. Lastly we provide a step-by-step validation of our generated patterns from the synthetic dataset.

Chapter 6 explains the detailed support to execute various tasks by the users interacting with the system. We discuss at-length how our standard graphical interface helps users conduct with their own spatio-temporal dataset easily and extract patterns from them. We also explore the reasoning behind different parts of our design and how they are designed.

Chapter 7 discusses the implementation strategies of our visualization and overall performance of our system. We discuss how the implementation choices we made reduces the initial time length to run a number of datasets. We then analyze the scalability of our system for different number of timestamps. We provide

two case studies with two types (numerical and image) of real-life datasets where we analyze geological trends visualization, and small change visualization.

Chapter 8 provides a detailed analysis of map projection and implementation of equal-area map projection with ContourDiff. We also compare the generated images from ContourDiff with direct numeric dataset as input and projected dataset as input to understand the impact of equal-area projection.

Chapter 9 provides a overall summary of our findings and contribution of the whole research. Also, we discuss several future research directions that can be extended from our work.

1.5 Declaration

Below is the list of publications and the works that are prepared for submission (with collaborators) from this thesis.

- **Zonayed Ahmed**, Michael Beyene, Debajyoti Mondal, Chanchal K. Roy, Christopher Dutchyn, and Kevin A. Schneider. ContourDiff: Revealing Differential Trends in Spatiotemporal Data. *International Conference Information Visualization (IV)*, 2021. (accepted and to be published as a full paper).
- Michael Beyene, **Zonayed Ahmed**, Chanchal K. Roy, Christopher Dutchyn, Kevin A. Schneider, and Debajyoti Mondal. ContourMove: Exploring Temporal Changes in Large Geo-spatial Data. Global Water Futures 2nd Annual Open Science Meeting, 2019. (presented as a poster).

Throughout the thesis, the term "we" refers to the author and the reader following the norm of traditional computer science writing style.

2 Literature Review

Visual analytics of spatio-temporal data is challenging since it requires harmonious integration of spatial relationship and time series. Moreover, the exploration techniques need to use appropriate visual mapping to allow users have correct interpretation of the information. In this chapter, we provide first a detailed overview of the characteristics of spatio-temporal data and different workflows to tackle various analytical problems. Then we review the literature spawned from exploration of spatio-temporal visualization. Lastly, we review the literature related to revealing differential trends in spatio-temporal data.

2.1 Visualization Workflow

Before we dive into the reviews of different strategies to visualize spatio-temporal data, we look into how visualization models work. These models can efficiently organize and manage temporal geographic data sets that are associated with additional attributes and semantics [160]. Zhong et al. [162] provided a complete workflow to realise spatio-temporal visualization from raw data, as shown in Figure 2.1.

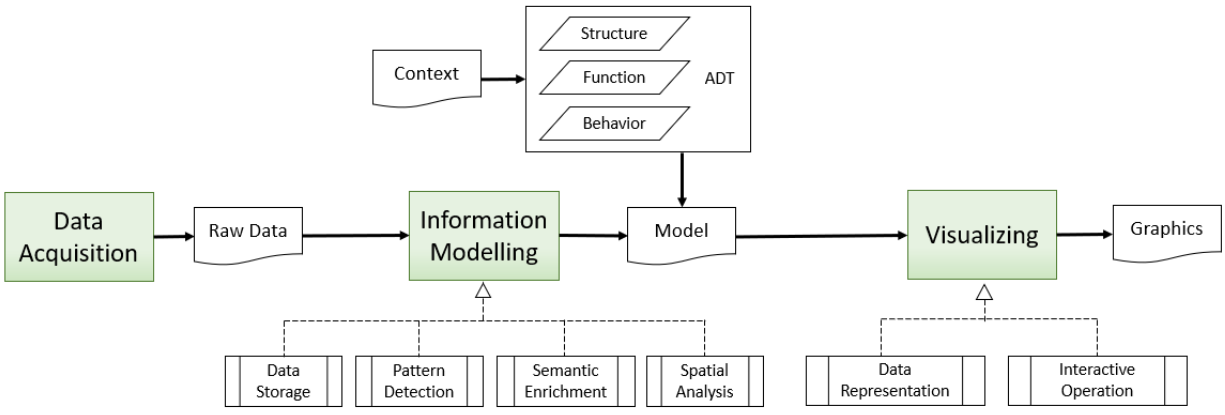

Figure 2.1: Overall workflow to realise spatiotemporal visualization reproduced from Zhong et.al. [162].

This is a general methodology of a visualization strategy that has also been adopted by our work in this thesis. We acquire raw matrix or image data and detect discernible patterns based on different characteristics from the data. These patterns provide semantic enrichment of the information that we process and later represent through our interactive visualization strategy.

2.1.1 Spatio-temporal Data

From a broader point of view, spatio-temporal data is a subset of ensemble data [117, 151]. *Ensemble Data* is a collection of multiple time-varying data sets (called *ensemble members*) that are generated by computational simulations of one or more state variables across space [117]. Wang et.al. [151] identified five orthogonal dimensions of ensemble data, i.e, **variable**, **location**, **time**, **member** and **ensemble**. While these five dimensions lead to five features of ensemble data, Spatio-temporal data is generated from *location* and *time* dimensions.

Spatio-temporal data involve three major components: space (where), time (when), and objects (what) [113]. Hence, a spatio-temporal phenomena can be classified according to their spatial or temporal properties. Spatial property can be classified into discrete or continuous; discrete phenomena occurs at distinct spatial locations whereas continuous phenomenon is defined everywhere over the territory (e.g., population density or air temperature) [96]. Temporal properties, i.e, the type of changes occurring over time in a spatio-temporal phenomenon can be classified further according to [24]:

- existential changes: appearing, disappearing, reviving of objects or/and relationships;
- changes of spatial properties of objects (location, size, shape);
- changes of thematic properties, i.e., values of attributes.

Sometimes only one type of change takes place or is of interest to an analyst, but in many cases one needs to consider several types simultaneously [18]. Hence, a reasonable solution in visualization and analysis of spatiotemporal data should offer, at least: resources for treating both the spatial and temporal dimensions (spatiality and temporality); domain independence (generality), freedom for the user to handle the visualized data and apply filters (flexibility); connection with several data sources in a practical and efficient manner (interoperability); and data mining based on spatiotemporal clustering (mining).

2.1.2 Data Type and Data Abstraction

The strategies to extract information from raw data depends on the type of data we are trying to explore. The spatio-temporal data visualizations are often designed to analyze uncertainties in simulation model dataset, explore parameter setting or extract interesting features [32]. Chen et. al. [32] gave a detailed taxonomy of data visualization techniques that includes visual design in simulation space (e.g., boxplot-based visual summarization, visual mapping, silhouette-based illustrative rendering, etc.), parameter space analysis (e.g., visual steering, parameter space projection, etc.) and data processing in feature space (e.g., data sampling, data reduction, data clustering, feature extraction, etc.). In this thesis, we mainly focus on simulation space and feature space analysis techniques since we are interested in finding trends in overall data pattern.

Another important data-related characterization is the number of time-dependant variables we would like to visualize at once. These variables can either be basic data types such as integers, real numbers, categorical

enumerations or advanced data types such as vector data and matrix data. Moreover, we can either represent data where each temporal primitive is associated with a single data value (i.e., *univariate* data) or multiple data values (i.e., *multivariate* data).

To provide users in-depth understanding of a situation, visualization models can either present data itself or its abstraction. Aigner et. al. [10] argued that analyzing larger datasets with visualization can get cluttered and overcrowded and hence, it is better to derive data abstraction reflecting needs and interests of the users. Calculating aggregated data, visualizing interesting features or event analysis in a time-series data are categorization of such abstractions [92, 107, 43].

One can look into spatio-temporal data visualization also through time dimension. In this perspective, the strategies can be divided into two ways: *static* and *dynamic*. Static representation means visualization in still images that does not change over time. On the other hand, dynamic representation utilizes physical dimension time to convey time dependency of the data [10]. While static image can be used to provide an overview of a full spatio-temporal dataset, it can also be linked with other smaller views to provide an in-depth analysis of the data itself. Dynamic representation, such as animation, storytelling, image sequence can help exploring movement in objects.

Another way to look into representing spatio-temporal data is through the type of tasks in temporal dimension. Andrienko et. al. [16] categorized the analysis tasks into the elementary tasks (e.g., given a moment, identifying spatial locations and objects) and the general tasks (e.g., comparing behaviors at the same or different time interval). They mainly apply the concept of searching into the three data components (time, space, objects) distinguished in the "when, where, what" framework of Peuquet [113]. Focusing only on the temporal component of spatio-temporal dataset, they deliver a four categories of targeted analytical tasks as following:

- describe characteristics of this object (location) at the given time moment;
- describe the situation at the given time moment;
- describe the dynamics of characteristics of this object (at this location) over time;
- describe the evolution of the overall situation over time.

In our thesis, we form our visualization strategy focusing on characteristics of an object (geospatial variable) over the whole spatial location over time.

2.1.3 Visual Task Analysis

Visual analytics combines visualization, data management, data analysis, and interactive exploration into a scalable, usable, and easy-to-operate system. While information visualization techniques is mainly motivated by information seeking mantra ("overview first, zoom/filter, details on demand") [129], Keim et. al. [79] adjusted the mantra to bring its focus toward visual analytics by including *Analysis* part:

"Analyse First - Show the Important - Zoom, Filter and Analyse Further - Details on Demand"

Moreover, there are several types of tasks that can be carried out by these visual analytics strategies. Wang et al. [151] provided a classification of six tasks that can be adapted to visualize ensemble datasets. These six tasks are:

- **Overview**: Provide a concise visual summary of ensemble data and convey the overall uncertainty.
- **Comparison**: Visually identify the difference between two ensembles (e.g., two ensembles of values at two spatial locations) using juxtaposition, superimposition, and explicit encoding.
- **Clustering**: Divide members or ensembles of objects into separate groups, where similar members or ensembles are in the same group.
- **Temporal Trend Analysis**: Reveal how a single member, some members, or all members evolve over time.
- **Feature Extraction**: Extract geometric (e.g., vortex, eddy) or topological (e.g., source, saddle, sink point) features from the uncertain field of ensemble data.
- **Parameter Analysis**: Build connections between ensemble data and the simulation parameters.

In this thesis, we provide a temporal trend analysis of geospatial variables over a period of time. We now explore different visualization strategies employed to visualize spatio-temporal data and the analytical task they carry out through the interactivity and multiple-linked views.

2.2 Summary-based Visualization

2.2.1 Statistical Summarization

The boxplot-based visual summarization approaches are mainly designed to get the trend aggregation results, such as a series of work based on a design named boxplot [100]. Boxplot is a graphical method for displaying five descriptive statistics: the median, the first and third quartiles, and the non-outlying minimum and maximum observations. It also gives a straightforward way to compare data sets.

Surface box plots [100, 75] generalize the standard box-plots in two and three-dimensions, where the surfaces are computed based on the three quartiles at each two-dimensional point. Surface box plots and various other extensions [76] have been proposed to portray the distribution of geophysical variables over a long period of time.

Potter et al. [115] proposed a hybrid summary plot for two-dimensional data distributions that incorporates a collection of descriptive statistics and density information. Figure 2.2A1 shows how this summary plot is designed. The abbreviated box plot shows the data distribution and the moment plot shows feature characteristics. The histogram calculates the density of distribution which is displayed using a symmetric

display and colormap. Finally, a distribution fitting allows the users to compare the data against a well known distribution [115]. The authors use this 2D summary plot to explore humidity and temperature across 28 altitude slices, as shown in Figure 2.2A2. Each slice visualizes joint mean, joint standard deviation, skew variance, and covariance with 1D summary plot along the axes for 2 variables.

Sun and Genton introduced functional boxplots [139] for functional data and provides case study for spatio-temporal preceipitation data. An interesting observation is that instead of having five summary statistics as in a classical boxplot, the functional boxplot has the envelope of the central 50 percent region, the median curve, and the maximum non-outlying envelope as descriptive statistics. Later, they adjusted this functional boxplot for correlations to perform functional and spatio-temporal data visualization and outlier detection [140].

Contour boxplot [155] provides a generalization of functional boxplots that builds on the notion of functional data depth to contours and demonstrate methods for displaying the resulting boxplots for two-dimensional simulation data in weather forecasting and computational fluid dynamics. Figure 2.2B displays contour boxplot with median as yellow curve, 50% interval as the darker blue band, 100% interval as the lighter blue band and outliers as dashed red curves.

Curve boxplot [103] proposes a generalized method to extend traditional whisker plots or boxplots to contours and curves traced in ensemble vector fields. Figure 2.2C shows this extension as the order statistics of 27 historic hurricane tracks originating in the Gulf of Mexico between 1920 — 2012 (left) and a curve boxplot visualization of an ensemble of 50 simulated hurricane tracks (right). Here the color scheme is similar as 2.2B, except that the outliers are presented with red solid curves.

While statistical summaries provide us with a deeper understanding of the characteristics of data at discrete locations, analysis that attempts to reveal a pattern over a geospatial region often relies on visualizations that draw trend-lines or blends color across the region of interest.

2.2.2 Isocontour

Field data such as data in scalar field, vector field, tensor field can be visualized using isoconoturs in 2D to reveal patterns and distribution of data [32]. These patterns are usually extracted before visualization.

A visualization system has been developed in [67] where contours are extracted from selected statistical properties (such as range, mean, median, maximum mode, standard deviation,variance, skewness, kurtosis) for each (x, y) position in 2D space, shown in Figure 2.3A.

ViSUS/CDAT [116] is another framework that provides a variety of techniques to visualize mean and standard deviation. Moreover, contours and height fields are used independently or in combination to indicate the ensemble estimate that in turn, provide a global understanding of the ensemble behavior. This is portrayed in Figure 2.3C.

Ensemble-Vis [117] provides a general framework where the authors display the variable mean through color mapping and the standard deviation using overlaid contours, as shown in Figure 2.3B. Rainbow colormap

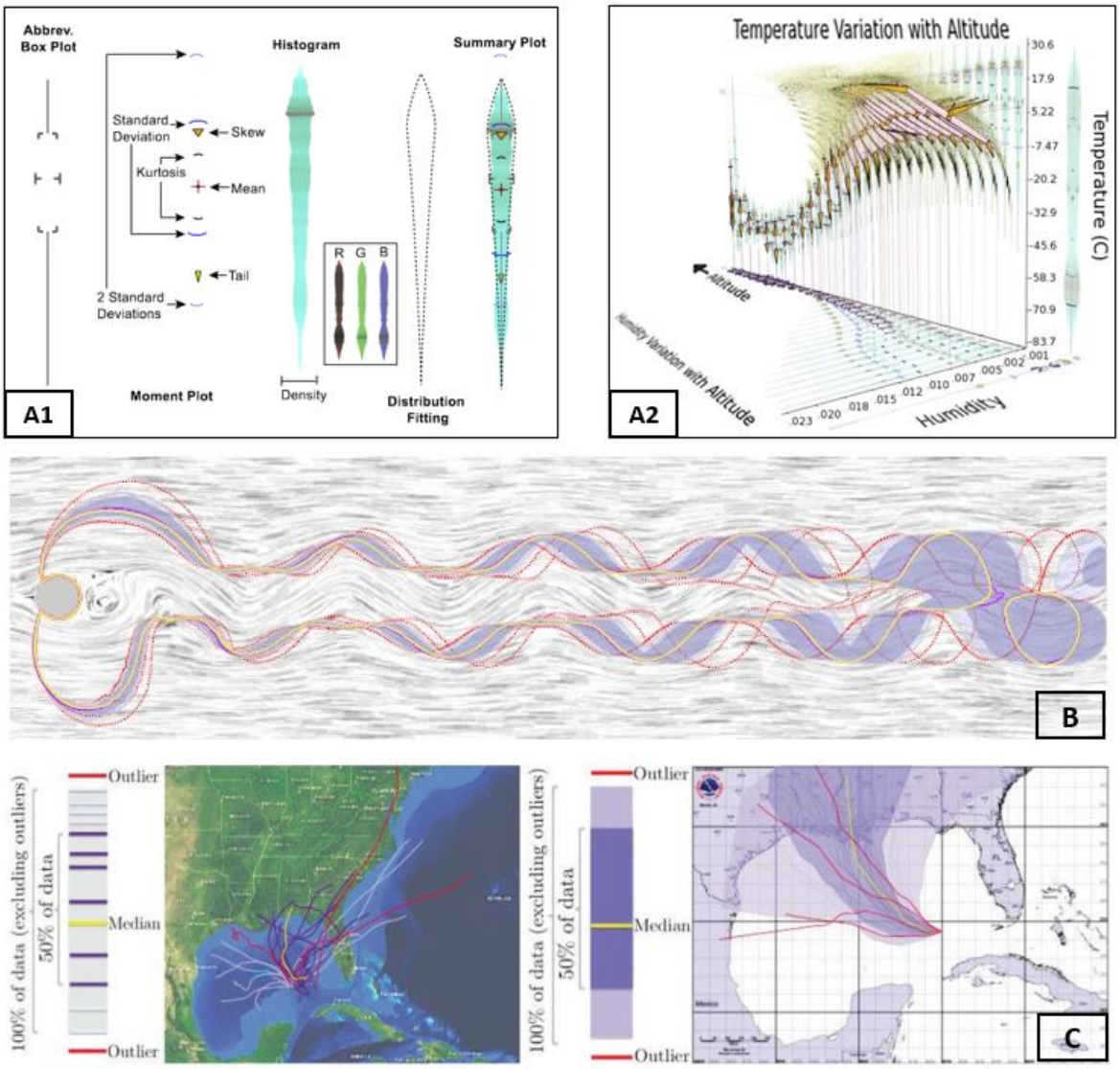

Figure 2.2: Boxplot based visualizations. (A1) Anatomy of the summary plot, (A2) 2D summary plot of temperature and humidity averaged across altitude slices [115] © 2010 IEEE; (B) Contour boxplot for an ensemble of the pressure field of a fluid flow simulation with a Line Integral Convolution background image as background [155] © 2013 IEEE; (C) Visualization of the order statistics of hurricane tracks (left) and a curve boxplot visualization (right) [103] © 2014 IEEE.

for temperature and relative humidity, and user-defined sequential colormaps are used to differentiate between isocontours. Spaghetti plot is another view added with this framework that consists of the isocontours chosen by a user-defined timestep, a variable and a contour value for that variable. These spaghetti plots [40] are named so because of its resemblance to a pile of spaghetti noodles, is a tool frequently used in meteorology to examine variations across the members of an ensemble over space.

Spaghetti plot with color encoding has been developed in Noodles [126] where a combination of three sequential color schemes with six hues has been used, as shown in Figure 2.3D. This framework also generates the spaghetti plot interactively which is useful in situations where the user is curious to see the spaghetti distribution at a specific location. However, as Bo Ma and Alireza Entezari [95] argued, even though the isovalues of isocontours are assigned different colors, overlaying a large number of isocontours brings significant clutters in Spaghetti plot. Hence, they propose to select isovalues of interest from a simplified Spaghetti plot that is displayed in 2.3F. This system renders a set of representative isocontours for multiple isovalues. For each isovalues, they chose a significant mode as representative and encode the variability of isocontours within each mode cluster to the contour width of its representative.

Concept of isosurfaces has been used by Fofonov et al. [46] for spatio-temporal multi-run data. Their earlier work allows for the comparative analysis of individual simulation runs over time within the ensemble and the inspection of the influence of simulation parameters. It is based on isosurface similarity between the fields of different time steps and different runs. This is portrayed in Figure 2.3E. A generalization of this isosurface approach to field approach to measure similarity between spatial fields has been introduced by the same authors in [45] for spatio-temporal data.

Geospatial data over time are often visualized using a series of contour plots and heatmaps. Such plots commonly use various color schemes to leverage humans' natural ability to differentiate colors [120]. A rich body of literature [125, 36] examines effective ways of visualizing geospatial data using colormaps. Color blending [159] widely appears in analyzing correlations among a pair of variables over a spatial domain.

None of the mentioned charts or visualizations can readily visualize differential trends. In addition, potential attempts to blend contour maps from different timestamps and overlay isolines, become unreadable due to visual clutter as the number of timestamps grows.

2.2.3 Storylines

Storylines are a static way to represent spatio-temporal data where series of events marked with data and time information are displayed. There are different time labels such as specific graphic variables and symbols that are used in maps indicating changes. Minard's map of Napoleon's march to Moscow [147] is a great example of this representation as it uses different variables in a temporal context: line width indicates army size; horizontal and vertical position denote latitude and longitude; colour is applied to identify the direction of the march; and temperature is indicated in terms of a graph along the timeline. Additional variables have been defined in the context of information visualization [135].

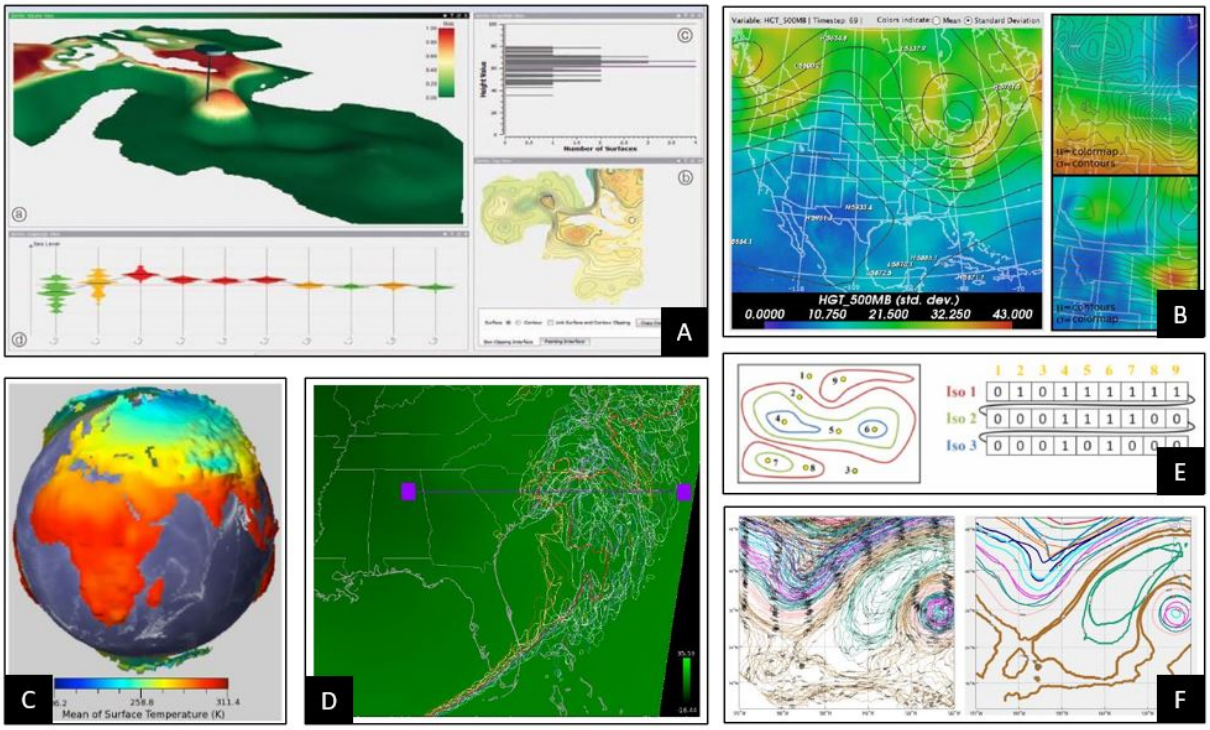

Figure 2.3: Isocontour based visualizations: (A) Exploration of ocean forecast consisting four views where the surface data is presented in (a) and (b), (a) indicates overlaid color-coded risk estimation over the region, (b) is a simple 2D top-down view that shows the mean surface of a specific time step, using iso contours and pseudo-coloring for the height field values [67] © 2013 IEEE; (B) Illustration of mean and standard deviation simultaneously using color and overlaid contours (left), closeups which both show the same region of data (right) [117] © 2009 IEEE; (C) Standard deviation is mapped to a height field and mean is color mapped [116] © 2009 IEEE; (D) Spaghetti plot of a temperature field with a colormap of mean values [126] © 2010 IEEE; (E) Using three isocontours to visualize local data differences between different data frames [46] © 2015 IEEE; (F) Comparison of Spaghetti plot (Left) and Simplified Spaghetti Plot (Right) for isovalue selection with isocontours from height data [95] © 2018 IEEE.

There have been many other efforts to create storylines, especially with movement data. For example, a movement analysis framework has been introduced in [13] that analyzes positions of 50 trucks along with their trip origins and destinations, and provides a number of information related to individual trips, clustering of trips, summarization of trips etc. by interactive visual analysis. Demsar and Virrantaus also explored spatio-temporal patterns in movement data by a novel approach, 3D space-time density of trajectories, which is a generalisation of standard 2D kernel density around 2D point data into 3D density around 3D polyline data (i.e. trajectories) [38]. They tested the approach using basic simulated data and real time movement data of vessel movement trajectories of ships in the Gulf of Finland.

Rinzivillo et al. [121] used progressive clustering approach on spatiotemporal movement data of cars to analyze their trajectories and plot them along streets on a 2D map. It is ideal for actors moving with uniform velocity and provides highly accurate geographic information on the movement. Tanahashi et al. [141] created storylines of different characters in movies such as Star Wars, The Matrix or Inception to show

how they interact at what point in time. But as [130] addresses, it only provides approximate location information based on the actors. Hence the authors presented a novel 2D visualization, *StoryGraph* [130], that provides an integrated view of time and location. It removes visual clutter, occlusion and difficulty to track time in common 3D charts. They also provided real life case studies that shows clear spatio-temporal patterns over a range of time.

2.2.4 Space-Time Cubes

Hagerstrand [56] first introduced a three-dimensional space-time model with time added as an additional spatial dimension. In its basic appearance, the cube has a base which is a representation of geography (along the x and y axis) and a height which represents time (along the z axis). The main advantage of the cube is that the whole evolution of a timeline can be shown within one image.

Kraak [83] integrated geo-visualization with Space-Time cube and extended its functionality with operations of changing viewing perspective, temporal focusing, and dynamic linking of maps with corresponding symbols. Again, Peuquet and Kraak [114] suggested that trajectories of object movement can be represented using the technique of space–time cube. This technique has also been used to detect changes in dynamic spatiotemporal social networks in SocioScape [119]. In this case, the base of the cube is the field of social interaction and the communities are indicated as spheres. Their horizontal position in the cube reflects their physical position at the time of the sighting and vertical position indicates the time of sighting. To indicate movement of a community, a line is drawn between two consecutive sightings. Turdukulov et al. [148] also used space-time cubes to create 3D event graphs and show the evolution of the tracked cloud objects by attribute values of the objects, by object paths and by events. In this case, the clouds are indicated as spheres with the radii proportional to a selected attribute of the cloud. Kapler and Wright also introduced similar 3D interactive unified geospatial and temporal analytical model called GeoTime [77] that is used for various geographic applications.

However, this approach does not scale well to a large number of samples due to the occlusion problem in 3D space [138]. Moreover Gatalsky et al. [48] compared 2D visualization with 3D time-space cube visualization; time-space cubes are comparably harder to understand for end-users and interpretation errors can be made. Another main limitation of this approach is that any movement along the vertical axis is lost since that is used for time [162]. That problem can somewhat be overcome using dynamic viewing strategies.

2.2.5 Time Maps

Time Maps are another great way to indicate changes in subsequent time frames. A number of literature proposes automated or user-controlled techniques related to time maps to indicate movement in a spatial location. As one class of dynamic representation of spatiotemporal datasets, timelines are used as a basis of mapping events over time and dynamic information is displayed as an image-series. Hewagamage et al. [63] used this representation with a combination of geographical map to show spatiotemporal events.

Another classic yet simple dynamic representation is a small set of changing maps only, in terms of static pictures that change over time. According to Tufte [146], the technique of map iteration or "small multiples" is a juxtaposition of several maps where each map shows the state of a phenomenon at a different time moment. Map iteration is available in SpaTemp [136] enabling unified and integrated processing of spatial, temporal and thematic data. A number of conventional cartographic methods such as lines connecting object positions at successive time moments, arrows indicating direction of movement, time labels when particular locations are visited are employed in this system to indicate location changes.

Slocum et al. [134] used this technique in MapTime that explores spatiotemporal datasets associated with point locations through animated maps in automatic or user-controlled frame rate. Data increases and decreases are represented by circles of two different colors, where circle size represents magnitudes of change. Guo et al. [55] also explored spatiotemporal visualization using "small multiples" approach where a small map for each time step is designed in which the cluster membership of the objects is represented by color.

2.2.6 Dynamic Movement

Representation of routes in static maps or multiple time maps can be advantageous because of its view of the whole trajectory made by an object. But while speed of an object is in consideration or we need to explore a convergence of different objects with respect to time, then static representation can have certain limitations. Moreover, separate plots for each timestamp or small multiple plots are difficult to explore when the number of timestamps is large. Map Animation can overcome these limitations.

Koussoulakou and Kraak [82] first experimented with animated maps and compared them with static maps. They reported that users can perceive more using animated maps. Andrienko et al. [14] proposed a time window technique for the animation of object movement, where route fragments passed by the moving objects are shown specifying time lengths. But this animation does not provide an opportunity to the analyst to compare states of the phenomenon under study at different time moments directly. Hence, the same authors proposed an integration of transformable time-series plot [15] dynamically linked to a map discovering interesting and significant features in data, as shown in 2.4A. Again, the authors introduced CommonGIS [17] that provides choropleth maps, time maps, and time aggregators with space-time cubes for supporting visual investigation and comparison of spatial development scenarios. This is displayed in 2.4C.

Tempo-Vis [9] also provides time-based interactive exploration using a time-slider by which users can navigate through different time points and explore social network activities and this is displayed in 2.4B. Again, a system named VIS-5D [64] animates perspective views of time series data that have three spatial dimensions, namely, latitude, longitude and altitude. There is another technique similar to animation called "fading" that is available in atlas of Switzerland [69]. In this technique, a map or image referring to one time moment gradually "fades" while a map or image of another time moment gradually appears.

One limitation of animation is that change blindness can cause people to not recognize minor changes [108]. Again, according to a study by Robertson et al. [122], animations can have presentation value but they are

not effective for analytical tasks due to limits in short term memory. Since animation or such interactive approaches are difficult to adapt for identifying subtle differences even between two consecutive windows, we do not use such methods in our thesis.

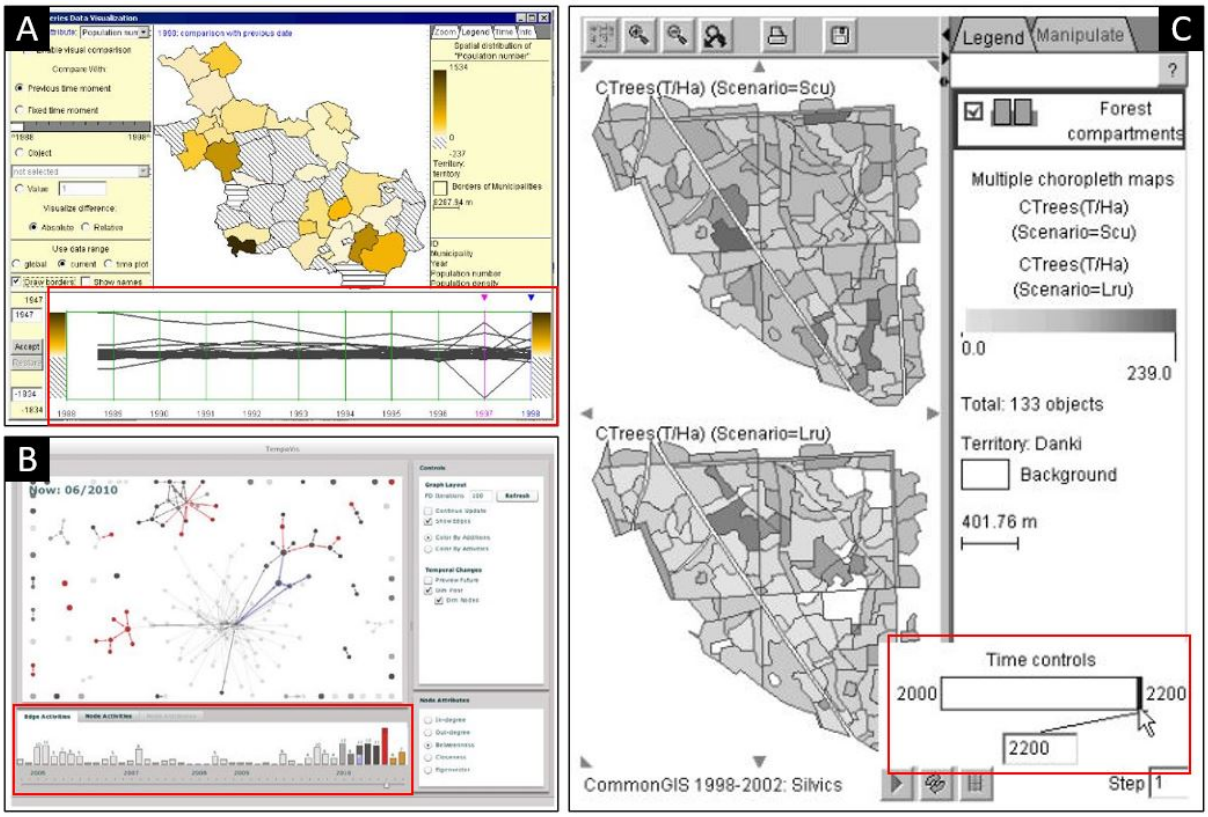

Figure 2.4: Time-series animation based visualizations: (A) Temporal trend of population change between 1997 and 1998 of two municipalities where the red box indicates momentary screenshot from animated movement of data [15] © 2019 Physica-Verlag; (B) TempoVis is equipped with a time-slider interface indicated in the red box that can be used to check monthly conversation statistics of a social community [9] © 2019 Springer-Verlag Berlin Heidelberg; (C) CommonGIS allows users, with a time control, to select time moment to compare statistics or run map animation for observing the evolution of spatial patterns over time [17] © 2003 IEEE.

2.2.7 Graph-based approaches

Graph-based approaches have also been explored to help users to navigate and explore time-variant data. TransGraph [54] provides a visual mapping of volumetric data into a 2D plane that abstracts data evolution over time in different levels of detail. Weber et al. [154] presented a new approach for the visualization of time-series data with periodic behavior using SpiralGraphs. ThemeRiver [59] visualizes thematic variations over time within a large collection of documents that have been used by [33] to represent the time series of normalized cluster centroids for 5 variables of a climate model data set. An interactive framework for the 3D visualization of the time-series of Web graphs has been proposed by Itoh et al. [71] where Intra-graph,

Inter-time, and Inter-facet relations were used enabling users to explore the evolution of time sequences.

2.3 Temporal Trend Analysis

Temporal trend analysis reveals how the data of interest evolve over time. This analysis can be done using a single member of the dataset or a collection of dataset. All the static and dynamic representations of time-varying dataset discussed earlier, especially, baseline, storylines, time maps or animation can easily visualize the changes through time. But here we take an in-depth look into systems or frameworks that readily capture trends of changes. Since we analyzed spatiotemporal data as a subset of ensemble data, we mainly describe the three approaches suggested by Wang et al. [151] to capture these temporal trends: (1) view compositions; (2) time-series plots and their variations; (3) pathline based visualizations.

View Compositions

A group of visualizations rely on view compositions to reveal temporal trends. View compositions can be of two types: Justaposition and Supercomposition.

- **Justaposition**: This approach utilizes side-by-side comparison to indicate the differences of temporal information for different time steps. For example, Wang et al. [152] presented Nested Parallel Coordinates Plot (NPCP) to get the temporal trend for 30 days of precipitation from Weather Research and Forecasting dataset. Domain scientists have also used high dimensional range query in authors' 30 small multiples to deduce correlation of different parameters from multi-resolution spatial temporal ensembles. Animation can also be considered a justaposition approach since it relies on users short term memory to build connection among several time steps [151]. Animation also juxtaposes views along temporal dimension rather than spatial dimension. For example, Hao et al. [57] used animation to display changes in 3D shape spaces comprised of plasma particles over time. They mainly use a slider to capture the time step of the current visible member item, cluster or pattern. Their system can also take input from csv files where each file corresponds to a time step in a member. Based on user request, the system can perform a number of temporal analysis, or it can load and visualize pre-calculated results.

- **Supercomposition**: This approach overlays the visualization results from different timestamps for comparison. While this can result in visual clutter, the approach provides a single view to users so that they need less mental attention to compare results over time. For example, Ferstl et al. [44] developed an approach to analyze spatio-temporal evolution in iso-contours obtained from ensemble weather predictions. They argued that connection between these contours can hardly be established with animation because huge amount of information needs to be memorized by users. Hence, they introduced space-time cluster surface where a number of isocontours are stacked on top of each other for a user-defined number of timestamps. This allows users to inspect each iso-contour in-detail and also

using a slider, inspect different temporal clusters. But this stacked representation can also introduce visual clutter while the number of timestamps is high.

Time-series Plots

Time-series plots can effectively summarize the temporal trend over a selected number of timestamps and demonstrate them as high level abstractions [151]. The *temporal view* section in the interface of EnsembleGraph [131] provides a visual summarization of all regions with similar behaviors over time. In the graph-based interface, vectors indicate the behavior pattern in each time frame and edges are temporal connectivity of different behavior patterns. plotKML [62] explored the changes in Land Surface temperature uttilizing time series where temperature values from ground stations over the spatial region can be interactively explored by clicking at points of interest over the time series. Hydrovise [72] is a open source, web-based software where the authors developed an evaluation module that allows users to conduct hydrologic model performance assessments for a selected time period. The authors also compared between two streamflow prediction models for soil moisture to indicate peak timing differences of the predictions. Ovis [68] is a framework for visual analysis of ocean forecast ensembles which also provides a time series view. This view helps visualize and compare temporal trends of statistical measures and uncertainty with violin plot glyphs. The cluster information of members at different timestamps can also be shown via time-series plots as shown in [73]. The authors bundled curves of the same cluster into a strip to design this time-series plot and show temporal evolution of transport variability in 2D flow fields. Time-series plots have also been used to find patterns in taxi trajectory data [138].

Pathline Demonstrations

Pathlines are another intuitive way to demonstrate underlying temporal trend in uncertainty of ensemble data. Cox et al. [35] used a bunch of pathlines to to show the predicted hurricane tracks. One pathline is used to indicate one trajectory of the hurricane. Overlay of all the pathlines shows the areas the hurricane can affect with different probability. But when the number of trajectories increase, this can quickly produce visual clutter. Hence, instead of directly using pathlines, different statistical measures can be presented such as uncertainty cones [2] and curve boxplots [103]. Since these approaches can be encoded directly into simulation field, they are intuitive in demonstrating temporal trend and uncertainty evolution [151]. But they cannot present details of ensemble members at a specific time step, due to lack of synchronization among pathlines at different time steps [88].

Temporal pattern analysis has been used to visualize medical data [43], law enforcement data [31], war logs [130], multivariate geospatial data [127], anomaly in spatiotemporal data [28] etc.

2.4 Single Integrated View vs Coordinated Multiple Views

A very interesting insight is found when we dissected the literature on spatio-temporal data representation. There can be one single view where the provided visualization, in one map or multiple maps, indicate the trend in the whole dataset. Again, coordinated multiple linked views are a combination of several small views, where each view indicate different characteristics of the dataset.

2.4.1 Single Integrated View

This type of approach provides static overview of the spatial location which can be beneficial for two reasons: (1) This type of representation can be printed out like an image, which later on, can be observed and analyzed; (2) End-users do not need to be attentive to multiple number of small screens to get the overall picture. But this strategy also has a few issues: (1) If someone visualizes all data rather than portraying significant information, then the single view will create visual clutter. Thus appropriate measures need to be taken to preprocess and filter the data based on necessity and significance; (2) Since a screen size is defined for visualizing the whole image, one must carefully design the interface so that users do not feel out-of-place while choosing different interactive options.

Several works in the literature employ this single view strategy - 2D and 3D. Shanbhag et al. [127] presented three 2D techniques, namely, wedges, slices and rings to visualize changing patterns of population data. They show increasing and decreasing trends of yearly dataset in a spatial location. The authors also provide interactive user interface with zooming option to detect the changes. But this strategy has two problems: (1) When the number of timestamps grow, the visualization does not remain effective; (2) When the polygon representing a spatial location is very small, even with zooming, it is very difficult to understand patterns for a large number of dataset. Andrienko et al. [14] proposed a method that superimposes time graphs on multiple 2D maps. But as Srestha et al. [130] suggested that the time graph may occlude the data points on the map view. In addition, events that happened at the same location but at different times may occlude each other. To address the problems of 2D integrated view, researchers have also proposed 3D integrated views. For example, previously described Space-Time cube [48] is a static representation of temporal trend analysis strategy. Tominski et al. [145] placed 3D icons and plotted time paths over 2D maps to better visualize event occurring at the same location over a period of time. Srestha et al. [130] argued that, though 3D integrated views do not have the problem of occlusion, it's difficult to align time data with location in 3D. Additionally, as the number of events grows, the scalability reduces. So they introduced storyline based 2D integrated visualization strategy, *StoryGraph* where spatial and temporal data does not occlude each other [130]. This strategy provides a big picture of an entire set of events in location and time with details displayed on demand.

2.4.2 Multiple Linked Views

Multiple linked views are used often in providing essential details that can not be contained in a single window. Wang et al. [151] argued that multiple linked views can overcome the information overload problem faced by a single integrated view. But analyzing spatio-temporal or ensemble dataset require a process of exploration among those views, as well as interaction such as brushing, linking that drive the exploration [151].

Potter et al. presented a framework, Ensemble-Vis [117] that combines multiple linked views to analyze ensemble data. Provided by spatial overview, temporal overview and statistical overview, the authors focus on both short term weather forecasting and long term climate modelling. Diehl et al. [39] presented VIDa (Visual Interactive Dashboard), which is a client server system with interactive web interface with an overview map and multiple linked windows such as date selector, variable selector, minimap timeline, webmap spatial filters, meteograms, curve-pattern selector, and operations. CommonGIS [19] is another example where the authors use multiple linked views such as choropleth maps, time maps, time aggregators to visualize spatio-temporal data. MapTime uses small multiples of *"change maps"* since this has two advantages: (1) static change maps that display the change over time in one of three forms: raw magnitude, percent, or rate of change; (2) They permit users to see changes that are not apparent in animation. But this *small multiple* strategy cannot accommodate small temporal data scale. As authors in [130] argued that the temporal data is cut into large chunks to avoid generating too many map views. SimVis [42] is another multi-view visualization framework that uses multiple views to view and interact with multi-dimensional flow data.

Previously described time-series plots such as EnsembleGraph [131], plotKML [62], Hydrovise [72] and Ovis [68] also employ multiple linked view techniques to explore temporal trends in spatiotemporal data.

Andrienko et al. [12] and Landesberger et al. [150] provided strategies for 3D multiple linked views. Andrienko et al. [12] linked three seperate views: 3D space time-cube, a 2D map view and a timeline view to analyze spatial, temporal and thematic components of mobile phone call dataset and flickr photo dataset. Landesberger et al. [150] combined geographic view with data locations and the dynamic categorical data view to find patterns in spatiotemporal people movement data and clustered weather data.

2.5 Directional Representations

Representation of geographical movement of events or a variable is very important in literature as they portray overall trend, pattern and correlations for spatiotemporal dataset. These movements can be visualized using two approaches: (1) Vector Mapping and (2) Glyphs. Vector mapping utilizes movement or flow of a certain dataset to place as arrows all over the map. On the other hand, a *glyph* is a symbol with a set of pre-attentive features mapped to the data variables to be visualized. Over a map of interest, the symbols are placed with varied feature set according to the underlying data value. Such features can be size, shape, color, hue, orientation, density etc.

2.5.1 Vector Mapping

Some geospatial data are inherently directional, e.g., such as wind flow and ocean current. Vector mapping is commonly used to visualize such geographic features [143].

For example, Auer et al. [21] used the gradient tensor field to visualize large scale trends on the background and strongly expressed features at their position as local visualization. They used earthquake and climate datasets to show the trends of change based on the most persistent maxima of their dataset. This is shown in Figure 2.5A, where wind flow is visualized with a number of icons. Kersting and Döllner [80] specified a visual mapping of vector data by scene graph and project rendered 2D textures onto georeference geometry such as terrain surfaces. Telea and Wijk [142] presented a clustering approach for a given vector field with arrow count and generates compact vector field visualization using a hierarchical simplification of vector data. The simplification is obtained by a similarity evaluation function and merging clusters of data. Instead of plain arrows to show the vector field, they use curved arrows, computed from streamlines of different cluster of data. They mainly emphasize on displaying a few icons to show main structure of the flow. Although this creates a clear representation, clustering approach often misses the details and outliers. A depiction of this visualization is shown in Figure 2.5B.

Another important goal is to enable users explore events over space and time to discover patterns and relationship in non-directional statistical data without any trajectory information. Kim et al. [81] proposed a continuous spatiotemporal functional representation for this kind of data using kernel density estimation. They use a simple representation of discrete spatiotemporal data as an array of spatial distributions over time. The authors perform 2D kernel density estimation on the data distribution to obtain continuous representations from discrete data. Then, they extract flow maps over a location from data of subsequent time steps using a gravity-based model. Finally flow maps are overlaid on a map to reveal movement patterns within a geographic space. A complete overview of their system is shown in Figure 2.5C. They perform this technique on four different dataset to explore temporal trends within them.

2.5.2 Glyph based Visualization

Uniformly spread glyph overlays are common in visualizing spatial datasets. There are various type of glyphs that are used to visualize flows from real life dataset.

Wittenbrink et al. [158] visualized uncertainty in an environmental dataset using a number of vector glyphs such as uncertainty glyphs or arrow glyphs. The glyph length or area was scaled based on the magnitude of their data points to show the direction of uncertainty for ocean current and wind dataset. Sanyal et. al. [126] used solid circular graduated glyphs over a regular grid to visualize ensemble uncertainty in meteorological data. Moreover, they proposed *graduated uncertainty glyphs* that encoded the deviation of each member from the mean of the whole dataset using visual cues of size and color, as shown in Figure 2.6A1. Thus, each member is depicted as a circle where the size represents the deviation. Then all the circles

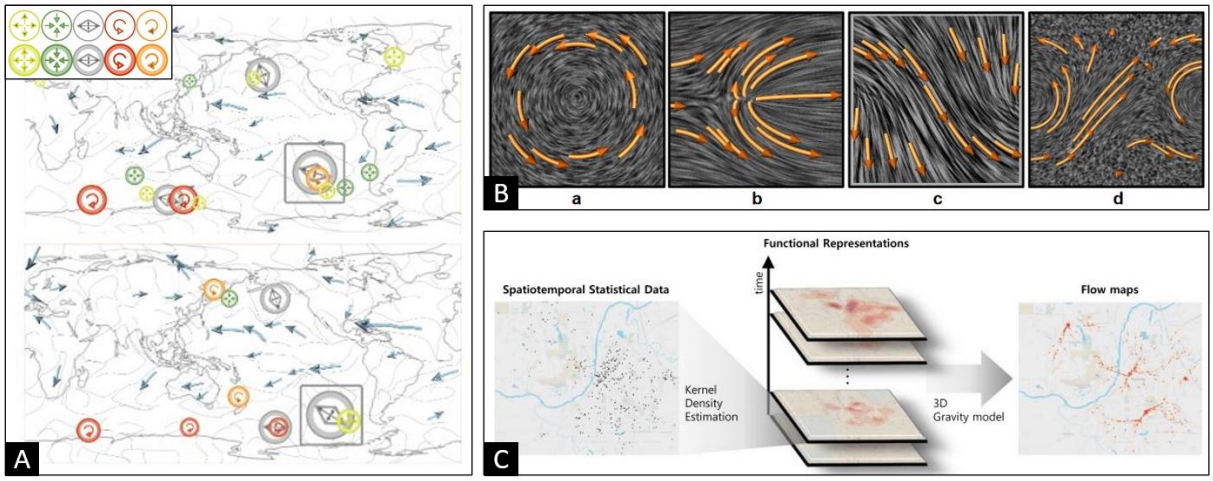

Figure 2.5: Isocontour based visualizations: (A) Time dependent simulation of wind dataset for two consecutive time stamps (top and bottom), where background flow visualization is large scale trends and icons are placed on top of that to highlight locations of strongly expressed field characteristics (all the icons are obtained from vectors and can be seen from the inset - top row: positive, and negative isotropic scale, shear, counterclockwise rotation, and clockwise rotation, bottom row: Respective extrema icons) [21] © 2013 IEEE; (B) Four examples of flow visualization where the background is spot noise and the vectors are shown as curved arrows [67] © 1999 IEEE; (C) The system overview shows how the discrete spatiotemporal data is represented as a continuous function (KDE), and then flow maps are extracted using 3D gravity model for temporal trends (movement flows) [81] © 2017 IEEE.

are overlaid at the same location, sorted size-wise. Smaller circles are shown with darker colors on top of bigger circles with lighter color. Figure 2.6A2 shows various types of graduated glyphs generated from their dataset. These glyphs are displayed over the entire grid in Figure 2.6A3 and along the contour lines in Figure 2.6A4. GlyphSea [101] is another interactive application that uses glyphs to encode the orientation of vectors in seismology and astrophysics data.

Hlawatsch et al. [65] used flow radar glyphs to visualize unsteady flow with static images. The strength of the presented glyphs is their ability to visualize time-dependent processes in a static way. Real-life case studies show that this strategy supports multi-scale visualization of flow data. Moreover, the glyphs show not only the distribution of uncertainty, but also directional information and its temporal behavior. Figure 2.7A1 and Figure 2.7A2 shows how the authors calculated flow radar glyphs. They portrayed each time-varying vector as a radial mapping of temporal variations in vector direction (Figure 2.7A1). The direction is mapped to the angle and time to the radius of a polar coordinate system. They also extended these flow radar glyphs to uncertain flow fields (Figure 2.7A2), where they added percentile of uncertainty as contour curves. The total area is therefore shaped as a radar glyph. Figure 2.7A3 shows the placement of these glyphs over an entire grid.

Kehrer et al. [78] used *superellipse glyphs* to represent various statistics of a member at each grid point. For example, median of the data is shown with color of the glyph. Moreover, overall size, upper glyph shape, and lower glyph shape represent the quartile ranges $q3 - q1$, $q3 - q2$ and $q2 - q1$ respectively. This scenario

Figure 2.6: Glyph based visualizations: (A1) Construction of concentric circular glyphs; (A2) Various types of graduated glyphs where the size of the circle indicates deviation from the overall mean (small core means few and large outliers while a wider core indicates more deviation within members); (A3) Graduated glyphs are shown over the whole grid; (A4) Graduated glyphs are shown along the contour of overall mean with a spaghetti plot [126] © 2010 IEEE.

is shown in Figure 2.7B. Jarema et al. [74] proposed *lobular glyphs* to support the comparative exploration of 2D vector-valued ensemble fields. The authors modelled the directional distributions at every grid point using mixtures of Gaussians and presented them using the glyphs over a user-selected region. Figure 2.7C1 shows *lobular glyphs* for various directional distributions. Two linked views of a wind dataset is shown in Figure 2.7C2 where color coded modality of distribution over an entire region is displayed and Figure 2.7C3 where directional distribution of rectangle marked region in 2.7C2 is displayed with glyphs.

Hölt et al. [67] proposed an interactive system with multiple linked views that uses glyphs to visualize the risk of sea depth in ocean surface simulation data. Figure 2.8A depicts this visualization clearly. A time step in overall distribution of an entire surface of Gulf of Mexico is shown in center and then two linked views are provided for further interactive visual analytics. The left view shows depth position histograms of the simulated sea surface. The right view is glyph representation of the data where the horizontal line corresponds to a chosen critical sea level, where each glyph's color depicts the risk corresponding to how much of the distribution is above that critical level. Liu et al. [89] extracted user-defined derived features (i.e. vortex) from vector field data and visualize them with glyphs (Figure 2.8B1). These glyphs are also used to compare monthly data, as shown in small multiples of Figure 2.8B2. For example, January month shows eight different group of vortex, encoded with various color. The vortex size is encoded as the radius on the top and vortex shape (Height - H, Width - W, Depth - D, Location - O) shown in the middle. The same vortexes are shown over a spatial region in Figure 2.8B1.

Although our visualization uses vectors to depict the relationships among different spatial locations, our context is different for the following reasons.

- First, the data points in our dataset do not need to be directional.
- Second, the magnitudes associated with the directions are of two types, increasing or decreasing, based on whether the associated location has a higher value than that of its neighbors.

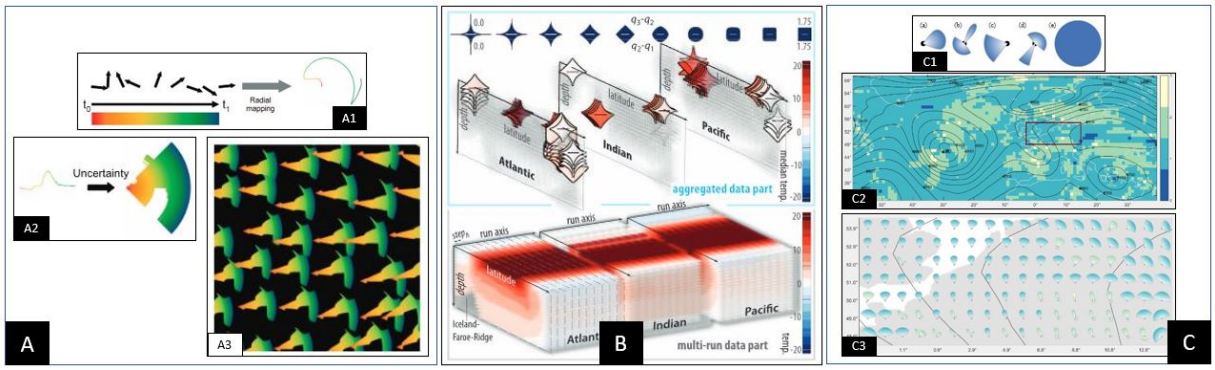

Figure 2.7: Glyph based visualizations: (A) A1 shows visualization of time-varying vector with radial mapping where the left sequences create the right glyph, with color mapping as an indication of temporal behavior of a variable, A2 shows extension of glyph with uncertainty area, A3 shows a simulation field with glyphs [65] © 2011 IEEE; (B) Multi-run data on 2D cross sections through the Atlantic, Indian, and Pacific ocean is shown in the bottom and various properties of glyphs are encoding aggregated properties of this dataset in the top [78] © 2010 IEEE; (C) C1 shows lobular glyphs for various directional distributions: (a) unimodal pdf, medium variation and (b) bimodal pdf, main mode (low variation), second mode (high variation), and (c)-(d)corresponding pie glyphs, (e) uniform pdf, C2 shows a region with distribution of wind dataset and C3 shows a zoomed-in view of rectangle marked region in C1 with proposed lobular glyphs [74] © 2015 IEEE.

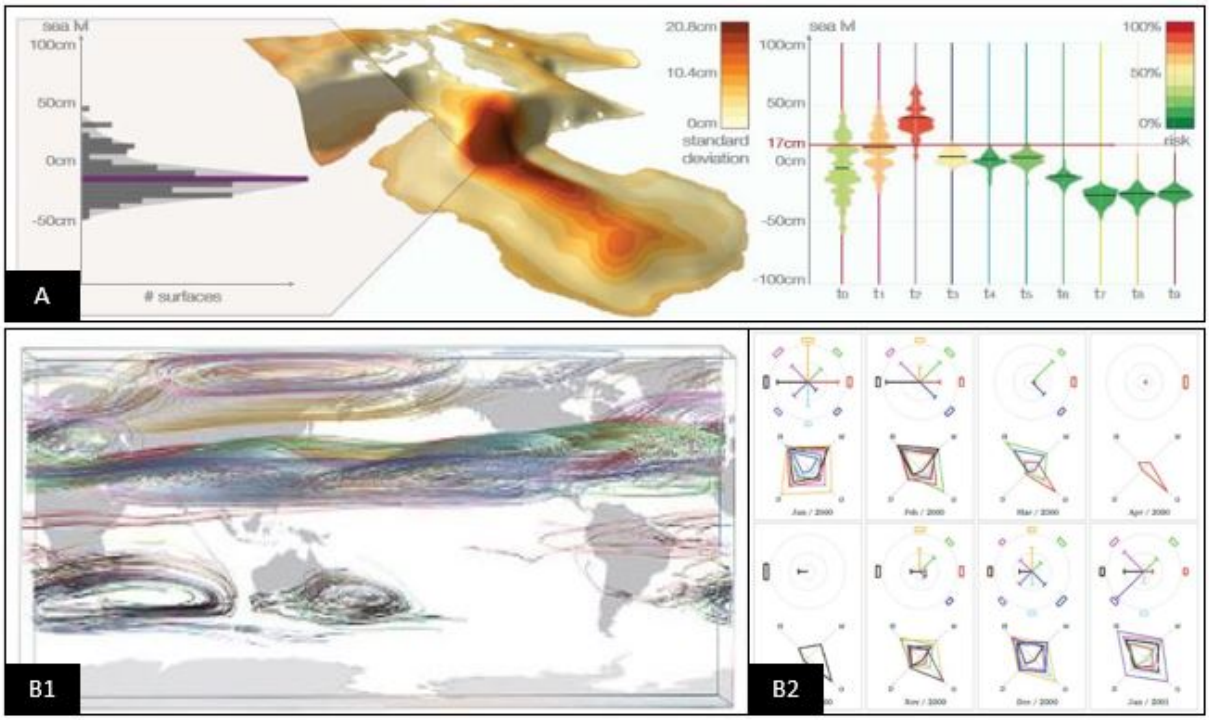

Figure 2.8: Glyph based visualizations: (A) Multiple linked views where ocean data of a timestamp is presented in the middle, histogram in the left and color coded glyphs in the right [67] © 2013 IEEE; (B1) Vortex glyphs of various groups are shown for one month over a spatial region, (B2) Small multiple of vortexes to compare monthly vortex data [89] © 2016 The Visualization Society of Japan.

- Third, the directions are not 'flows', rather they depict historical relations among different spatial locations.

2.6 Summary

In this chapter, we introduce various visualization approaches to show changes in spatiotemporal data. We discuss summary-based visualization (boxplot, isocontour, spaghetti plot, color overlay), static representation of changes (storylines, space-time cubes, time maps), dynamic movement of time frame, graph-based approaches to display how data evolves over time. Then we describe various approaches to capture this evolving nature as an analysis of temporal trends (view compositions, time-series plots, pathline based visualization). We explore single integrated view and coordinated multiple views of trend representation. Finally, we analyze Directional representations at length with vector mapping of geospatial data and glyph overlay to visualize various flow data. We also indicate contextual difference of our research from other vector or glyph representation of spatiotemporal data.

3 Dataset Description

In this chapter, we explore various dataset that we use to validate our approach to visualize spatio-temporal dataset and their trends. Our approach can be adopted to any spatio-temporal dataset, which is spatially continuous. So, we use both real life meteorological dataset and image dataset to prove that. Moreover, we utilize a synthetic dataset to explore the readability of our visualization. Here we explain all three types of dataset.

3.1 Real-Life Meteorological Dataset

There are a number of real life meteorological spatio-temporal dataset to use for visualization and validating a system. Here, we use Weather Research and Forecasting (WRF) model [133] output as a case study. WRF has been one of the most used and steadily growing atmospheric model for research and numerical weather prediction with over 36,000 registered users and over 1,340 peer reviewed publications [118]. This model uses Advanced Research WRF (ARW) dynamic solvers and rich suite of physics packages under a finer spatial and temporal resolution to calculate and collect weather data over the whole world. Hence, this model has been used as a numerical weather prediction system used for reproducing local weather and climate at high spatial resolutions.

Our model domain of the dataset is composed of 699×639 grid points with 4km horizontal resolution. Also the region it covers is the western Canada, from British Columbia and the Yukon to the West and the MRB and the SRB to the East. This region covers 2560 km in the North-South direction and 2800 km in the East-West direction. Figure 3.1 shows this region.

The observed dataset contains data values related to 26 geospatial variables from January 2013 to September 2015 monthly over western Canada. We used different climatic parameters (monthly data from 2013 to 2015) to investigate the performance of the system and analyze their historical differential trends.

3.2 Real Life Image Dataset

Since our proposed system can also work on images as dataset, we use timelapse images to evaluate our system. Here, images can be transformed into a positive matrices, each corresponding to a single timestamp. The dataset is related to various human motions for different actions where the motion of the whole body is captured. We try to capture and visualize small changes between almost identical images with these two

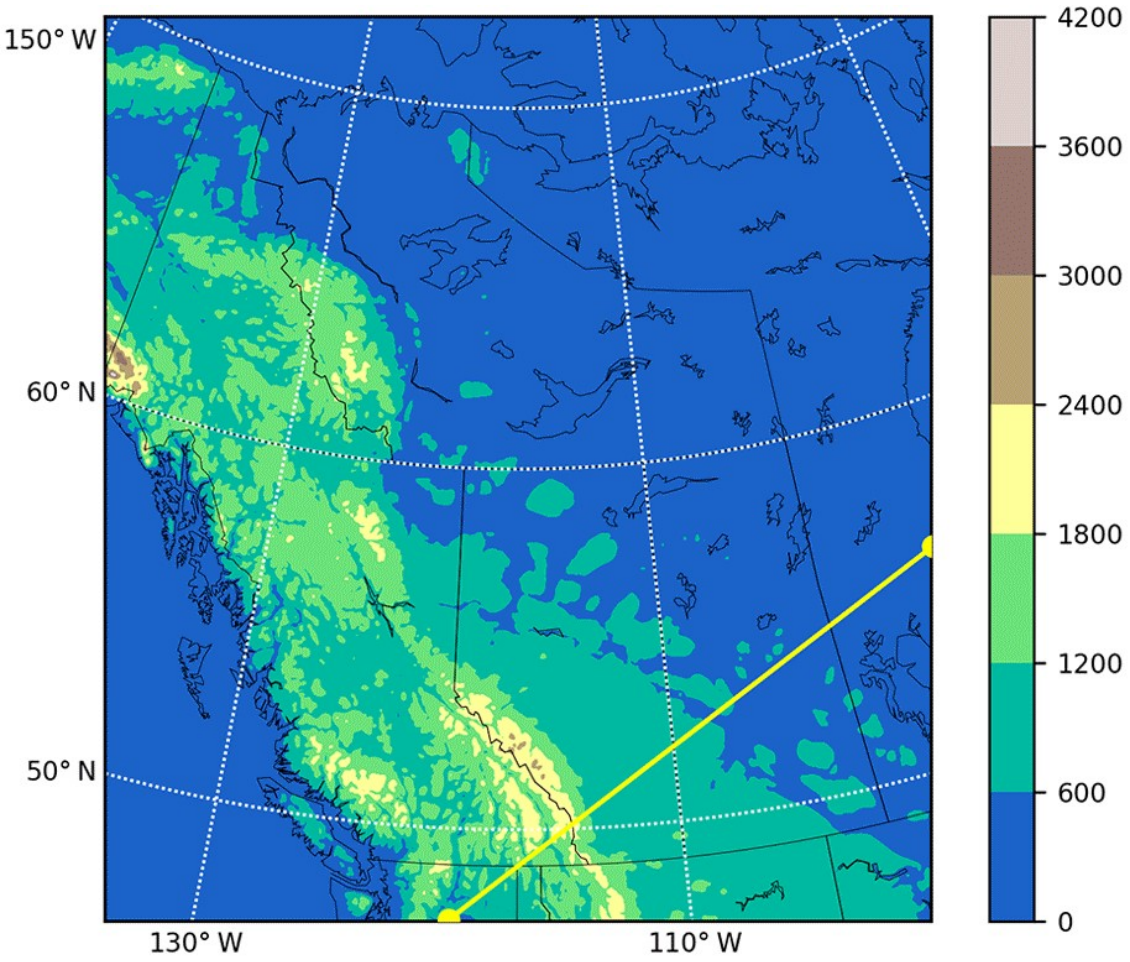

Figure 3.1: Elevation Map of Western Canada, licensed under CC BY 4.0 [84].

datasets (e.g., Figure 7.7).

3.3 Synthetic Dataset

We also generate a synthetic dataset to evaluate our system and let users understand the process of readability of our visualization.

3.3.1 Objective

Synthetic data can be described as the data that is generated in a simulated system, performing simulated actions. Real-life geospatial data variables can change all over the map, maintaining patterns or trends. Thus, it can be very difficult to identify and understand these complex patterns. Moreover, we also need to validate our system step-by-step so that the readers can understand how a ContourDiff visualization is created. We keep this synthetic dataset simple to demonstrate the readability of our created visualization. This also allows us to reveal the strengths and weaknesses of our system at a more granular level.

3.3.2 Design

We generate synthetic dataset by leveraging the idea of a general elevation map and how contours are generated from this. We explain contour plot and how we compute them in the next chapter. But here in image 3.2, we see how elevation of a mountain in 3-dimension can be converted into a 2-dimension figure.

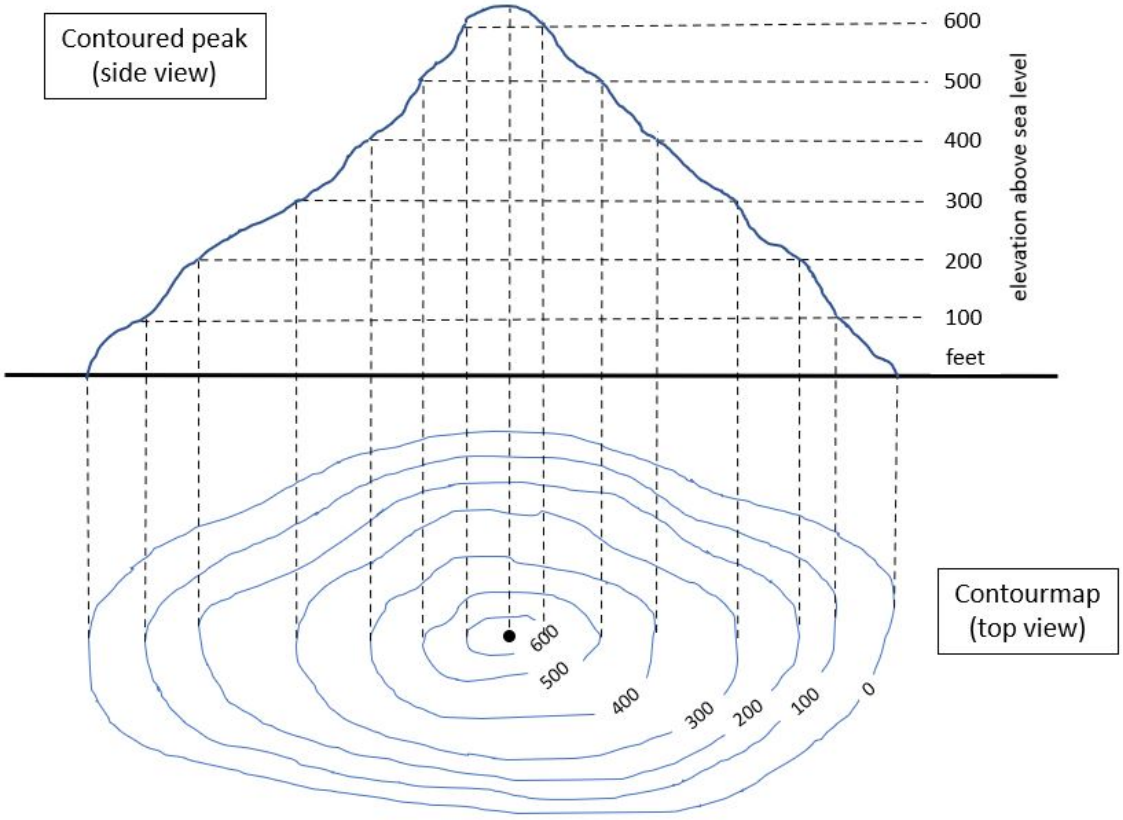

Figure 3.2: 2D Contour Map created from a elevation of a mountain.

We can see how the value of elevation of the map decreases from the center towards the border. So the value at the center is the highest. On the other hand, an elevation of a valley has the lowest value at the center and gradually increasing towards the boundary. We utilize this idea to generate our own synthetic dataset.

3.3.3 Implementation

Here we discuss the implementation of our synthetic data:

- We construct a sample classification dataset by taking a two-dimensional standard normal distribution and defining classes separated by nested concentric annulus such that roughly equal numbers of samples are in each class [4]. This was implemented by using python's *makegaussianquantile()* function. A total of 40,000 samples have been used to populate the total area, each containing 2 features, namely a (x,

y) coordinate and a value v. We assign the values such that most nested class obtains higher values.

- We define 4 classes since that follows our isoline thresholding mechanism to create three isolines on the base map. To create contour lines as the concentric spheres, we check the number of samples from innermost to outermost sphere. We do this because to create multiple contour lines, we need to create the inner most contour line first identifying the class so that the value of each sample in that class is not updated while calculating other classes. We simply check for the number of samples in each class first from inside and then if the class has more than one sample, we define all the samples in that class with similar value.

- The dataset was then interpolated and reshaped into position using python's *np.interp* and *.reshape()* functions respectively, into a 200 $\times$ 200 matrix, respectively, such that the point density plot for each cluster takes the shape of a peak or valley.

- The covariance of the dataset is fixed to ensure no difference in sparsity. Moreover, all the contour lines are wiggly, but to better represent real life lines, we do not apply any smoothing operation to smoothen the contour lines.

- We keep all the sample coordinates and their corresponding values in a file which we use to progress with our mechanism.

3.4 Differences Among Data Types

There are two significant differences between real life meteorological data and synthetic data - density and position of the contour lines. We control the density distribution of the contour lines in our synthetic dataset but it can be vary widely in real-life dataset. Again, our synthetic dataset is simple since we use this for the readability of our system but we see our real life dataset can be very complex.

Numerical data and image data are also different in terms of their processing in the system. We extract the geospatial variables that we use from the WRF dataset and normalize them. For the image data, we capture different image frames from video and later preprocess them before using to find patterns. This preprocessing is described in Chapter 5.

4 Overview of Initial Visualization Strategies

This chapter discusses a few strategies that we implemented while gathering initial understanding for spatiotemporal dataset and how to extract possible patterns. Moreover, we try to employ general techniques to visualize the data and find the challenges associated with them. Later on, we describe the possible solutions to overcome them.

4.1 Contour Overlay and Side-by-Side Plots

The driving question of our research has been whether differential trends can be visualized effectively over the spatial domain using a vector based approach with respect to contour plotting of a region. To explore that, we first discuss contour plotting in detail and how we construct the contour map for a location. Then we analyze the spaghetti plotting and how we blended colors to perceive a better understanding of the final plotting.

4.1.1 Contour Plot

A contour plot is considered as a graphical representation of a 3-dimensional surface on a 2-dimensional plane, which is typically made from continuous spatial data, e.g., temperature data all over Canada. For a value for z (e.g. temperature), isolines connect the (x,y) coordinates attaining that z value. The isolines can be thought of as slices of x-y planes along the z-axis. The contour map looks like a number of x-y slices stacked on top of each other. These types of plots are used in making topological maps. A common example would be an elevation map over a region of interest as we showed in 3.2.

We compute our contour polygons using Marching Square Algorithm [99]. It operates on a 2D grid whose cells' corners contain scalar values. The goal of marching squares is to classify cells using the values at their four respective corners. This algorithm is relatively simple and extremely fast. We will introduce the python library that has been used to implement the algorithm in our Implementation chapter.

Figure 4.1 shows a contour plot of soil moisture data extracted from Weather Reseach and Forecasting dataset over Western Canada in March 2015. While this provides us a basic skeleton about the different regions, shading will provide far more distinction. For shading, we can use different colormaps on the generated contourmap.

Moreover, these generated contourmaps vary on the data distribution. So, the dataset can be divided in different isocontour level to generate the whole map. The contourmap produced in Figure 4.1 has three

isocontour levels (0.2, 0.5, 0.75) which divide the whole region in four sections. Since we can't understand these regions here, we try to blend a colormap with the generated contourmap.

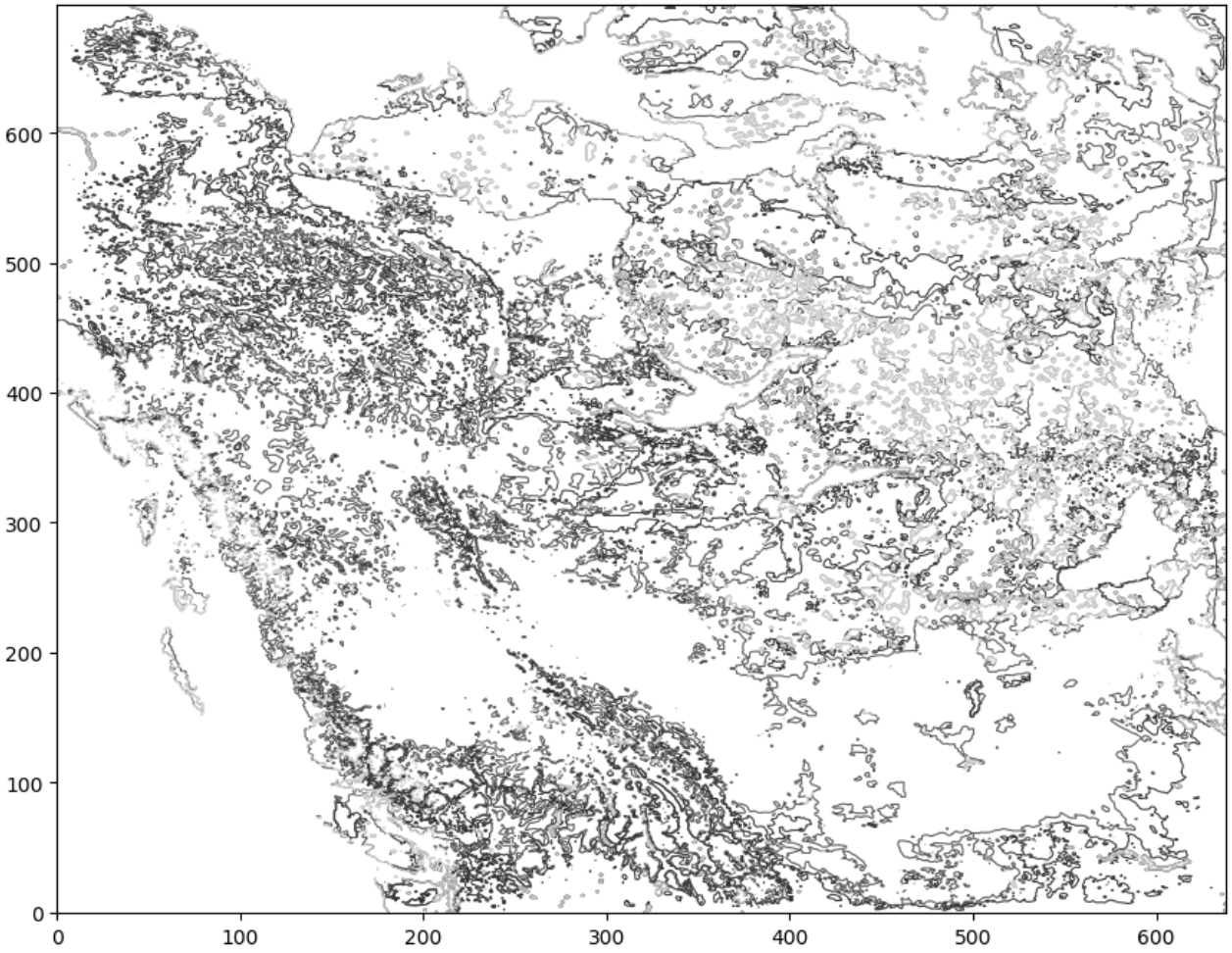

Figure 4.1: A contourmap designed by Marching Square algorithm.

4.1.2 Blending Colormaps

Blending colors is a common way to visualize information, e.g., temperature and pressure over a map. There are multiple blending modes to do so: each gives a different look to the underlying set of colors. Visualizing information for multivariate data can be challenging because this takes multiple variables to be displayed into account. For example, if we want to display both temperature and pressure in a colormap, a distinctive approach is needed to differentiate between two variables. While this becomes easier for univarite spatio-temporal data, still caution needs to be taken to utilize color, saturation and hue efficiently. Blending colormaps with conoturplots is a great way to deliver users information about different regions with different values. We blend a basic colormap with our contourmap from Figure 4.1 and the resultant image is showed in Figure 4.2.

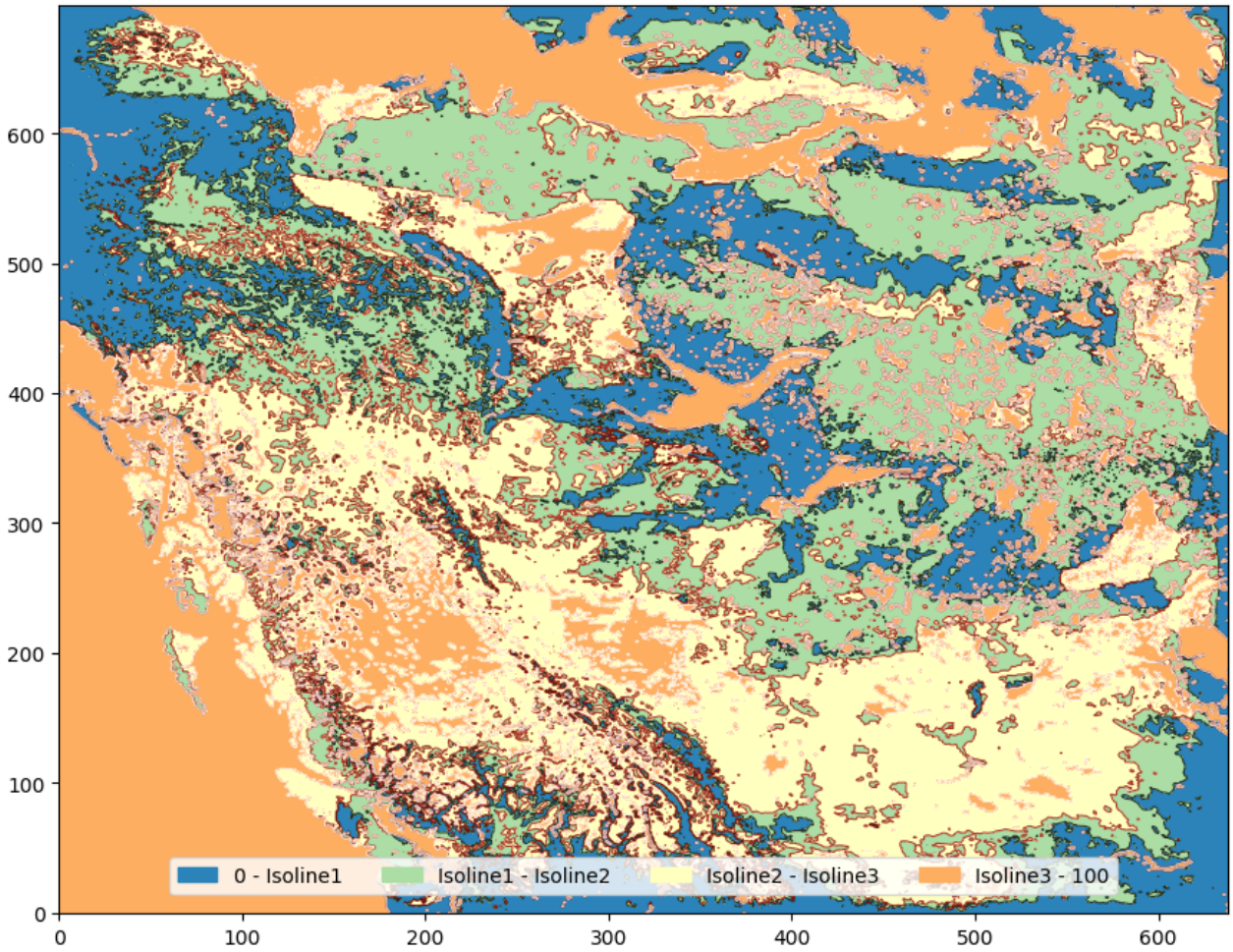

Figure 4.2: A color blended contourplot designed from Figure 4.1.

This basic colormap is shown as a legend on top of the map. Since Figure 4.1 has three isocontour levels, we choose different colors for mapping values of these levels. As the isoline values are $0.2, 0.5, 0.75$, color '*blue*' indicates the part of dataset having distribution range $(0-0.2)$. Likewise, colors '*light green*', '*yellow*', '*orange*' indicate the parts of similar dataset having distribution ranges $(0.2-0.5)$, $(0.5-0.75)$, and $(0.75-1)$ respectively.

4.1.3 Side-by-side Plotting

In an initial attempt to understand the nature of the problem, we examined the challenges of overlaying the contour lines and blending colormaps for different timestamps with real-life datasets. We can describe this as simply as picking up the difference in one's image now and after ten years. Or a simple *puzzle* game where the user needs to locate all the differences between two images to advance the level.

These approaches can be effective for two cases:

- Side-by-side colored contourmap tests our cognitive ability to acknowledge the finer detail in the images.

Now generally our brain can differentiate between images of a pair of different timestamps. For example, in the initial levels of the *puzzle* game, there can be only two images where you have to locate all the differences. So, to get all of them, you just need to carefully scan through the first photo and then compare that with the second one.

- Comparison becomes easier when there is very large variation between images of two timestamps. Decrease in detail puts less pressure on observation power and thus human brain can capture most of the information introduced in the images. For example, initial levels of the *puzzle* game can include two images with large differences. Moreover, with only two timestamps, variation can only come in binary form: increasing or decreasing, larger or smaller, thick or thin, etc. This binary choice availability of pattern finding increases the chances of being accurate and thus can be efficient.

On the other hand, the limitation of this approach includes:

- When the number of time samples increase, the high variability in contour lines makes them unreadable. Moreover, with the growing number of timestamps, there is high strain on human eyes. Since we have to locate the difference between each subsequent timestamps, it can lead to loss of information regarding overall change pattern of geospatial variables (e.g., see Figure 4.3).

- Another major problem with side-by-side plotting is that small, yet significant patterns often go undetected (e.g., see Figure 4.4). For example, the *puzzle* game may include two images with differences in finer detail that can be easily missed in naked human eye.

4.2 Direct Vector Overlay

4.2.1 Introduction

Since side-by-side plotting has not been proved useful, we try an alternate approach by combining all the information from different timestamps. While spaghetti plot can overlay contour lines for different timestamps on a basemap, this provides an unclear image where the overall difference can be comprehended. Meteorologists use spaghetti plot to portray uncertainty in ensemble forecasts in variables such as pressure, temperature, precipitation etc. Because ensemble forecasts naturally diverge as the days progress, the projected locations of meteorological features will spread further apart. Therefore the contours follow a recognizable pattern through the sequence. But if the number of timestamp grows or the contours get entangled, it becomes difficult to put confidence in the overall pattern. So, we discard overlaying contours directly to create a spaghetti plot.

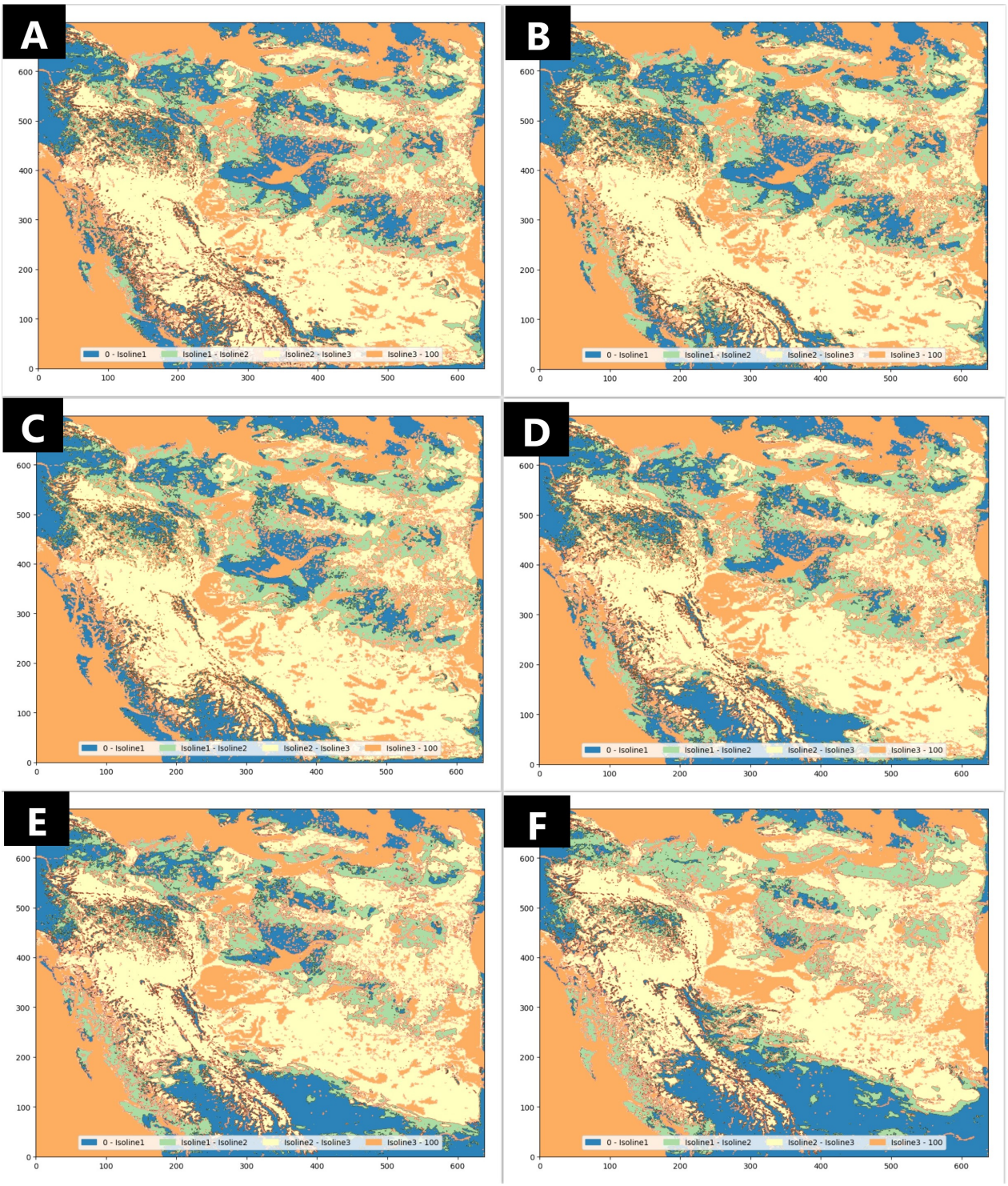

Figure 4.3: Contourmaps of Soil Moisture from WRF dataset over Western Canada for six consecutive timestamps: (A) February 1, (B) February 15, (C) March 1, (D) March 15, (E) April 1, and (F) April 15 of 2015 with isocontour values (0.2, 0.4, 0.75).

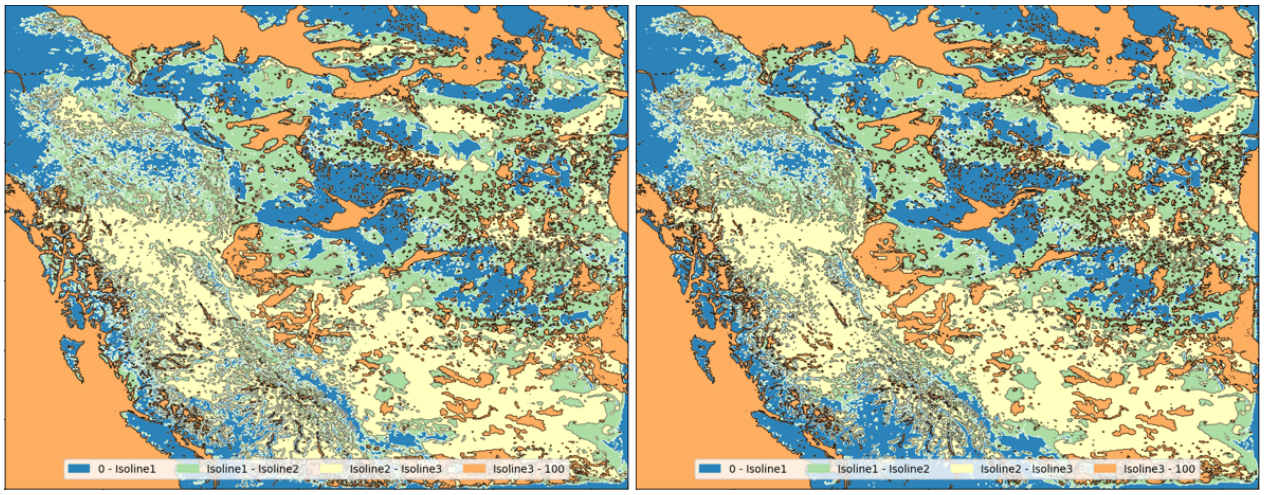

Figure 4.4: Side-by-side contour plots of soil moisture for two different timestamps, March 1, 2013 (left) and March 1, 2014 (right) with isocontour values (0.25, 0.5, 0.75). Here, it is difficult to perceive an overall understanding of the differential trends through cognitive reasoning.

4.2.2 Proposed Approach: ContourMove

We visualize the change information through vectors. A vector V is a directed line segment that has two components: magnitude m, the length of the vector and direction, the flow of the vector. We are interested in this since we can portray the temporal change information through magnitude and direction of the change over the spatial map through the flow. Moreover, our differential trend calculation at a spatial location takes the value difference with its neighbours at all angles. Since we are considering the location of the neighbours with respect to a location point, P, we also consider the tail of the vector at P and head to each individual neighboring point. Additionally, we can indicate the value difference between P and its neighbours with the magnitude of the vector.

We are interested in visualizing historical trends over a spatial location. In other words, we process the movement information from any spatio-temporal dataset and map that information over the spatial map. So we propose a novel approach ContourMove that creates a contour plot from the recent most timestamp and then computes a vector field for the whole map. But since we are only interested in the movement of a geospatial variable along the contour lines, we overlay that information over the colored contourmap.

Step 1: Creating ContourMap

We create a basic colored contourmap from the recent most timestamp t, so that it can create an overall structure of the geographic location based on a weather variable such as snow, temperature, precipitation etc. In this approach, we use a basic sequential colormap to create the contourmap.

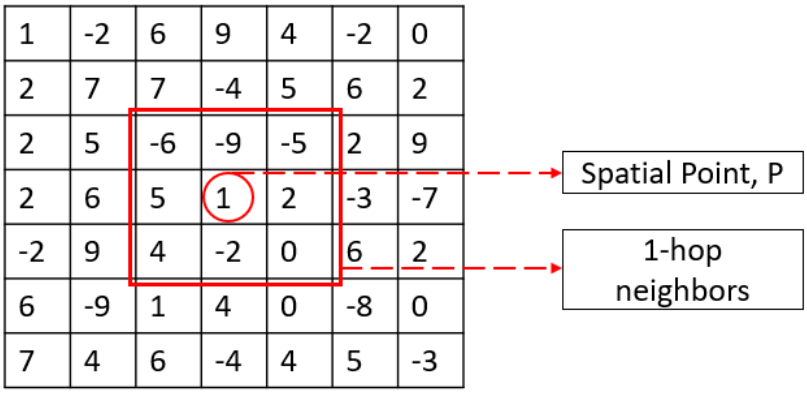

Figure 4.5: The 1-hop neighboring points are indicated for spatial data point.

Step 2: Computing Vectors

Next we compute the vectors from the data points at each spatial location. Following steps were taken to compute the feature vectors:

- We compared a spatial point's value with the value of its 1-hop neighbours in all directions. We define *hop* as the number of rows and number of columns surrounding the location. So, we consider 1 row and 1 column of data points as neighbours in this case. Figure 4.5 illustrates this situation.

- Then we calculate the scalar difference between spatial point P and its neighbours P_i. If the data value at spatial point P is x and the neighbours have values x_i, then the scalar difference at each spatial location P_i will be

$$|X(P_i)| = value(X) - value(P_i); i = 0, \ldots, 8 \tag{4.1}$$

- We compute the vectors from the scalar values and the respective position of the neighboring points. Let $|X(P_i)|$ be the scalar difference between X and a neighboring cell P_i, where $i = 0, \ldots, 8$. Then the x and y components of the vector $\vec{V_X}(P_i)$ is computed as:

$$(|X(P_i)|cos\theta(P_i), |X(P_i)|sin\theta(P_i)),$$

  where $\theta(P_i)$ is the angle identified in the neighbor selection step. We explain this step in detail in Section 4 when we compute vectors from 2-hop neighbours.

- We aggregate all the generated vectors for timestamps 0,...,$t-1$ over the contour map according to their own direction. So, all the directional vectors at all timestamps for one spatial location are added to get an aggregated vector. So, for one spatial point, we get eight aggregated vectors. Thus we compute the aggregated vectors for all data points.

Step 3: Vector Overlay

Finally, we overlay all the vectors on top of the created contourmap. But since there are eight vectors for all the data points, even for one small spatial location, it can become very noisy. Thus it gets difficult to understand the overall pattern even for one timestamp, let alone several ones. So, we filter the vectors based on the positions of contour lines. Thus it becomes more clear and presentable. Moreover, it shows a pattern of change along the contour lines.

4.2.3 Advantages and Limitations

For a single timestamp, this approach seemed promising over contour line overlays and color blending. Since contourmaps or color blending do not provide differential information of a spatial location, Contourmove provides a significant benefit compared to that. Figure 4.6 illustrates the trend of change in neighboring data. This provides an overall presentation of how the data values change along the contour lines. Moreover, the zoomed images provides a clear sense that not all data points have vector flow in the same direction.

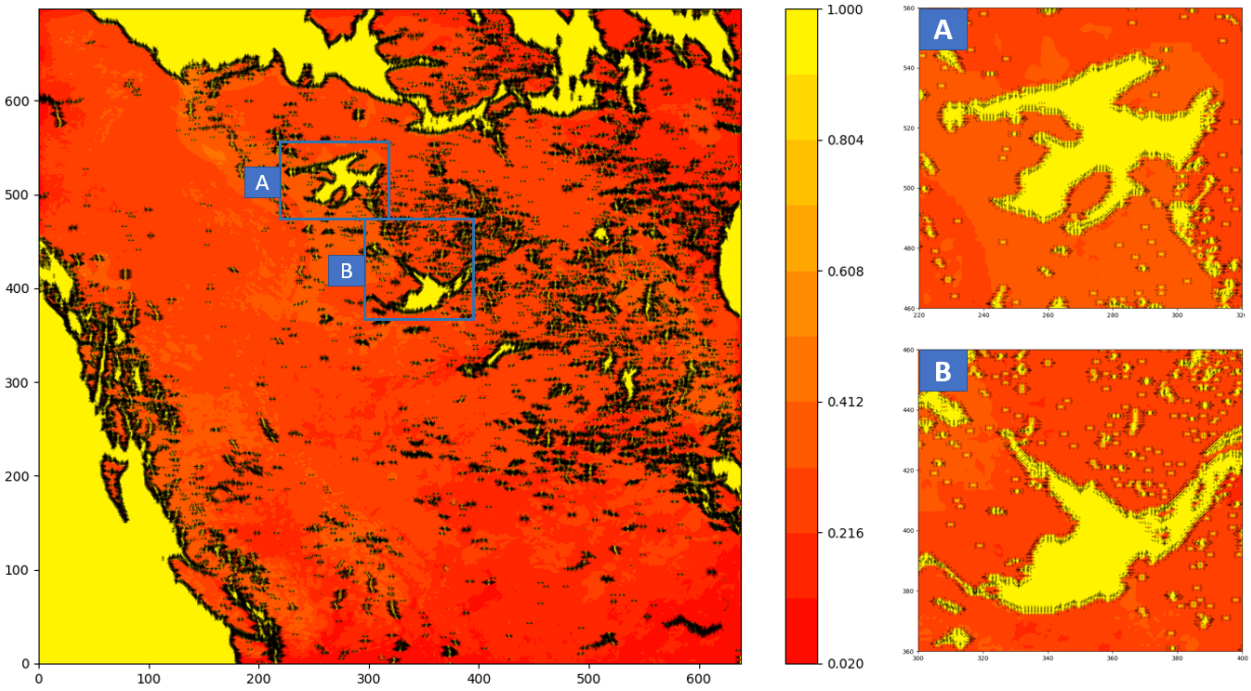

Figure 4.6: An illustration of Contourmove for September 1, 2015. A and B indicates two zoomed portion of the overall visualization.

Although for a single timestamp, this approach seemed promising over contour line overlays and color blending, it created clusters of vectors all over the map as the timestamp increases, e.g., see Figure 4.7.

4.2.4 Key Insights

We noticed several challenges associated with this initial vector representation:

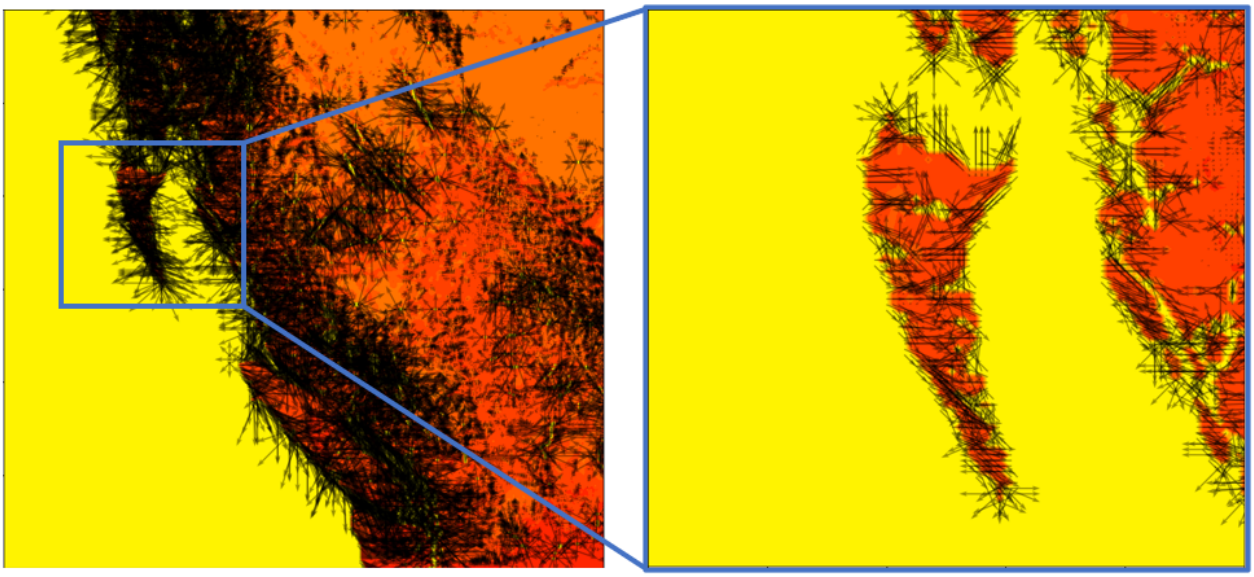

Figure 4.7: Direct vector overlay creates a cluttered representation of vectors over the contour map. This is a zoomed-in illustration of ContourMove for March 2013, March 2014 and March 2015.

Vector Crossing

We know that the edges alongside the geographic area are not smooth. So, the isolines generated from contourmap are wiggly and staggered. Thus the vectors that are generated from this approach can cross themselves. Even if the image is zoomed in to visualize a small 100×100 grid, the vectors are not clearly visible, tangled, and overlapped. This suggests that the vector rendering should be adaptive to the spatial resolution to avoid overlapping.

Random Directional Vector

The vectors generated from a spatio-temporal dataset depend on the differences between a spatial point and its neighbours. Now these dataset can create vectors with random directions while they get noisy. Since an aggregation over all the timestamps is representated by vectors, a miscalculation or wrong vector coefficient can create mismatched pattern. This suggests that a reasonably larger neighborhood should be chosen. More intuitively, we differentiate among the neighboring values based on their variation from the spatial point P. If the values at the neighboring members have higher value, then we call it *increasing* and neighboring members with lower value than P is associated with *decreasing*.

Pattern Mismatch

The changes may be increasing or decreasing and spanning over the whole region under consideration. Now this creates similar problem as looking at side-by-side comparison. Because without any discernible pattern, it is very difficult to understand the overall trend of change in the whole geospatial region. Moreover, eight vectors at a data point seems to be quite excessive in terms of human cognitive understanding. This indicates

that appropriate measures need to be taken to create a flow of information along the contour lines that can show the direction of change in a spatial location.

Low Interactivity

Slow computation of both contour and vectors posed difficulty to interact with the visualization. So, we need several implementation upgrades that will ensure smooth information retrieval with the visualization. Moreover, a highly interactive system is warranted to get a detailed understanding of the nature of data and deal with different visualization properties.

4.3 Modifications

We considered several modifications in light of our preliminary observations, which are briefly outlined here and explained in details in the Chapter 5.

4.3.1 Modeling Contour Lines via Graphs

We consider simplifying the contour lines to reduce the wiggling pattern. We also consider modeling the contour map using a graph with bend points as nodes and segments as edges. One reason for using such graphs is to speed up data processing, and the other is to quickly compute the significance of a segment after polygon simplification from the smaller segments that it replaced.

4.3.2 Vector Modification

We consider increasing hop number by 1 to extend the relation between a data point and its neighbours. The rationale behind this is to reduce the noise, and thus capturing consistent trends in the data. To reveal the local maxima and minima, we consider categorizing the vectors into two groups corresponding to the increasing and decreasing trends, and then aggregating them separately. Thus each location will have two vectors associated to it. Other reasons for this consideration are to obtain a better understanding of the data by separately examining these two types of differential trends, and also to make the visualization easily explainable to the inexperienced users.

5 ContourDiff

We discussed various challenges of direct vector plotting over a contourmap in the previous chapter. In this chapter, we will describe techniques to overcome those challenges using different approaches. We will provide a in-detail description of a novel, vector-based visualization framework, ContourDiff that portrays the changes of values of a spatial domain in different timestamps. The workflow in Figure 5.1 illustrates an overview of various steps of our approach. ContourDiff integrates a number of different ideas: contour map simplification, weighted graph creation, vector calculation based on the relation of a point with its neighboring points, vector crossing reduction, and quadtree based sampling to reduce vector clutter.

A ContourDiff generated image has two parts: (1) Background Plot, which is a contourmap that indicates our region of interest and (2) Feature Plot, which is the vector map plotted on top of the contour lines. While both the parts have a few similarities to the ContourMove approach that we discussed in the previous section, ContourDiff provides a far more in-depth understanding of change pattern in spatio-temporal data. Moreover, a single, static view of the overall trend analysis provides us a smoother and quicker interaction capabilities. In this chapter, we explain how they work together by progressively building the final output.

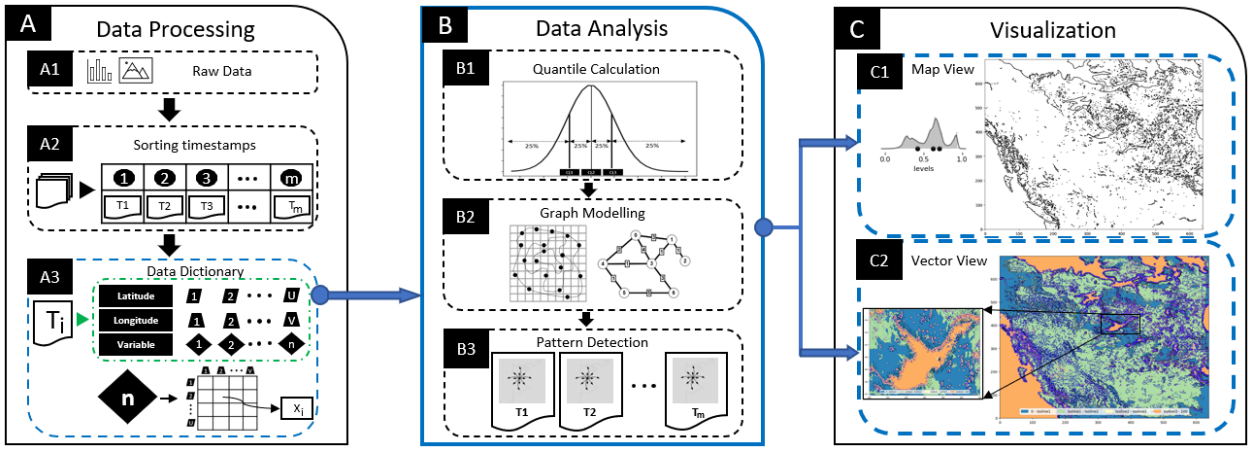

Figure 5.1: A complete illustration for the workflow to compute a ContourDiff visualization.

5.1 Processing the Dataset

Let N be the number of timestamps in the dataset. We preprocess each timestamp, T_1, T_2, ..., T_N to create a primary structure that facilitates the computation of the background contour map and overlaid vectors.

5.1.1 Raw Data

As we previously specified, ContourDiff can use both numerical data or image data to create a final visualization. We create a matrix structure from both type of datasets. A tabular dataset consists of spatio-temporal information has spatial location of a geographical area that is represented by its latitudinal and longitudinal position. The dataset also consists of geospatial information mapped with these positions. A geospatial variable V is usually represented as an one-dimensional non-negative matrix over the whole area. On the other hand, image data consists of spatial location via a pixel position. If a pixel position has (x, y) coordinates, then x is considered as longitudinal position and y is considered as latitudinal position.

5.1.2 Sorting Timestamps

Based on the collection of raw data, the temporal information can be addressed based on the collectors mindset. So the data can be collected daily, monthly or yearly. But for the processing of our data, we divide and sort them based on timestamps. If there are file inputs f_1, f_2, ..., f_N for N timestamps, T_1, T_2, ..., T_N, we sort them as such that timestamp T_1 occurs before timestamp T_2, timestamp T_2 occurs before timestamp T_3 and so on. We do this to process all the files according to their timeline as it is important to find the contour map of the last timestamp.

5.1.3 Processing Raw Data

Here we discuss how we preprocess both types of raw data - numerical and image. Numerical data is taken as .csv files of different timestamps. Since we use the Weather Research and Forecasting model for trend analysis and the dataset is 699×639 size, we also convert all numerical values in 699×639 matrix structure. For the image data, we also resize them in the same manner. Then we process them further in the following manner:

- First we do Otsu thresholding [110] on each image so that it outputs a binary image with a foreground and background.
- Then we perform watershed transformation [23] to separate the foreground and background so that we get the interesting regions of an image.
- Then we label the background zero to keep the foreground only.
- Then We capture the biggest polygon from the whole image.
- Finally, we create a dataframe consisting of latitude, longitude and pixel value.

5.1.4 Data Dictionary

We create a data dictionary to process both spatial and temporal information with regard to its geospatial variable. After sorting the timestamps, we create a tuple (x, y, v) for each timestamp T_i. While x and y are the screen coordinates computed from the latitude and longitude information, v is the normalized value accessed from the range of the variable. This normalization is necessary because we do not know the overall data distribution of different user input, hence data can have varying scales. Moreover, the algorithm we are developing does make any assumptions about the dataset like k-nearest neighbors and artificial neural networks algorithms do. So, the data is normalized using a standard normalization approach to scale it to a fixed range, from 0 to 1. The following equation has been used for this purpose:

$$\frac{(V - \max(V))}{(\max(V) - \min(V))} \tag{5.1}$$

Also it does not impact the final goal of visualization because it just changes the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

5.2 Computing the Background Plot

For the background of our visualization, we choose contour plot of the most recent timestamp. A contour map is generated based on the normalized values of a geospatial variable of interest. The reason for using such a contour map is that it provides an intuitive idea for the underlying data distribution using different quantile values (which is common in geospatial data analysis [100, 75]). The latest timestamp is chosen to visualize the historical differential trends with respect to the present-day situation.

5.2.1 ContourMap Generation

While we provided a detailed discussion about the algorithm that we use to develop a contourmap, we discuss here the process of generating that contourmap from the dataset. We take the dataset of the latest timestamp and check the overall data distribution. Based on that, we use statistical measure based on percentile of data to plot the contour map. We primarily choose three isoline thresholds: lower quartile (25^{th} percentile), median (50^{th} percentile) and upper quartile (75^{th} percentile) as the default isoline thresholds. Thus they create three levels of the dataset and they divide the whole dataset and effectively the whole region in four classes.

One reason to choose the quartile is that it mimics the standard box plot, as they are a standardized way of displaying the distribution of data based on a five number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum"). Another reason is that it places the contour lines sufficiently far apart so that the vectors don't overlap much over the contour lines. After rendering the contour map for the latest timestamp, we overlay the vectors computed from this and all the other timestamps on top

of the generated map. Note that the users can choose to change the isolines based on the data distribution visualized in our interface.

5.2.2 Initial Graph Creation

We construct a graph G from the previously generated contour map by extracting information from the contour lines. The graph $G = (V, E)$ is a set of vertices $V = v_1, v_2, v_3, ..., v_n$ and a set of edges $E = e_1, e_2, e_3, ..., e_n$. Each vertex v_i is defined as a collection of four inputs from a dataset at latest timestamp, T_n. This collection can be described as $[l, p, x, y]$ where l indicates all levels (isolines) of the map, p indicates all paths for a single isoline, x and y indicate coordinates of a point on a polygonal path. We describe the adjacency of two vertices v_i and v_{i+1} if they appear consecutively on the same polygonal path. We sorted the data dictionary and transformed all the data points of a geographical area in a sequential order firstly based on levels and paths. So, each data structure has 4 levels indicated by isolines in the map but each level can have different number of paths based on data properties and isoline choice. Hence, the overall graph structure is actually a combination of a lot smaller graphs generated from data points on different paths. Edges can be described as the connection between two consecutive vertices.

Later, this graph structure has been used to compute the vectors consistently considering the topological properties of the isolines.

5.2.3 Weight Calculation

For a fixed contour threshold, we obtain a set of polygonal paths. But as we can understand from a geographical structure, there can be lot of small paths based on data characteristics. While we do not know the number of such paths generated from a dataset, we understand that these small paths add visual clutter to the overall ContourDiff visualization. Hence we remove very small paths at each level. Moreover, we compute weights to indicate the importance of each path. The algorithm of computing the weight can be described in following steps.

- We assign the path with one nodes with 0 weight, hence they are automatically discarded from the visualization.

- For all the remaining paths, we first calculate the euclidean distance of all the subsequent vertices. In the Euclidean plane, let vertex v_i has (x_1, y_1) coordinates and a subsequent vertex v_{i+1} has (x_2, y_2) coordinates. Then the distance between the points is calculated by the following equation:

$$d(v_i, v_{i+1}) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2)} \tag{5.2}$$

  This distance is used as the weight of an edge.

- We aggregate all the weights of the connected edges on a path to get the final weight, w of a path p. So, if the path p has a subset of edges $E_1 = (e_1, e_2, e_3, ..., e_n)$ and the edges have weights $(w_1, w_2, w_3, ..., w_n)$

respectively, then the overall weight of that path, $P_1(w)$ can be computed by the following equation.

$$P_1(w) = \sum_{i=1}^{n} w_i \tag{5.3}$$

- Finally, we normalize the weights between 0 to 1 for all isoline component in all levels. We use the standard normalizing equation that we used earlier to preprocess the dataset.

We use these normalized weights to design the weighted graph.

5.3 Feature Visualization

Recall that we represent the characteristics of a geospatial region using vectors overlays on the contour lines. We use vector magnitude and direction to depict the differential relationship of a data point with its neighbors. To compute the vectors, for each spatial location s in a timestamp, we compute the differences for the value associated to s with that of its k-hop neighbors, $k \in \{1, 2\}$. For $k \leq 2$ and T timestamps, we obtain $24T$ vectors for each data point. We then group the vectors in two groups based on whether s has a smaller value than its neighbors or not. Finally, we aggregate the vectors in these two groups separately to obtain the increasing and decreasing differential trends at s.

5.3.1 Neighbor Selection

Given a geospatial variable, we first create a $M \times N$ matrix to represent the spatial region. We use the relation of 1- and 2-hop spatial neighbors with each data point to generate the vectors. We compute the difference in values of each cell X in the matrix against its 24 neighboring cells. Since we define 'k-hop' as k number of rows and columns around a data point, 2-hop means we use 2 rows and 2 columns around each data point, this process can be seen as a 5×5 filtering for each point.

For 1-hop, the positions P_0 to P_7 are identified counter-clockwise to identify the vector directions. For 2-hop, we consider the positions in two steps. The first step identifies points P_8 to P_{15} counter clockwise that are 45° apart from each other, where the first cell P_8 make a 45° angle with x-axis. The second step identifies points P_{16} to P_{23} clockwise that are 45° apart from each other in groups of two, where the first cell, P_{16} is 22.5° apart from x-axis.

5.3.2 Computing scalar and vector difference

For every location, we first compute the rate of change $\frac{dv}{ds}$, where v is a function that varies over the spatial domain and ds corresponds to the change in the spatial domain. We then categorize them into increasing and decreasing changes. Finally, we aggregate them separately to form two representative vectors. We explained in Chapter 1 how we define these rate of changes or differential trends as a generalization of image gradients and define them as *contour gradients*. Figure 5.2 shows how increasing and decreasing trends are calculated

for a spatial point (x, y). Though this utilizes 1-hop neighbors as the point's local neighborhood, ContourDiff utilizes 2-hop neighbors and their respective positions in consideration while computing overall trend vectors.

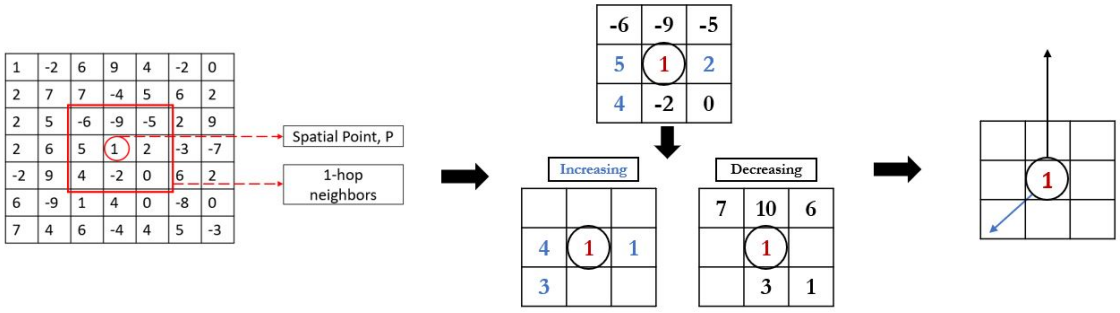

Figure 5.2: Computation of Contour Gradients for a spatial point with 1-hop local neighborhood

We leverage the matrix computation libraries to gain computational speed, hence formulated the vector computation process using shifted matrices. To compute the vectors for the 1-hop neighbors, we pad the $M \times N$ matrix with 1 row and 1 column of zeroes and create a $(M+2) \times (N+2)$ matrix. We shift the matrix in all eight directions to obtain an $M \times N$ matrix and then subtract the original matrix from the shifted matrix. For 2-hop neighbors, we pad the original matrix with 2 rows and 2 columns of zeroes and create a $(M+4) \times (N+4)$ matrix. But we shift this matrix in 16 directions to find $M \times N$ matrix in each direction. Here we explain the calculation with respect to a single data point.

Let $|X(P_i)|$ be the absolute difference between X and a neighboring cell P_i, where $i = 0, \ldots, 23$. Then the x and y components of the vector $\vec{V_X}(P_i)$ is computed as:

$$(|X(P_i)|cos\theta(P_i), |X(P_i)|sin\theta(P_i)),$$

where $\theta(P_i)$ is the angle identified in the neighbor selection step.

We see the components depend on the scalar difference and angle of each position with respect to X. In the first quadrant, both x and y components are positive. In the second quadrant, x is negative and y is positive. In the third quadrant, both are negative. In the fourth quadrant, x is positive and y is negative. we carefully investigate the angles of each position based on their quadrants. For first quadrant, we take the angle, θ as it is. For second quadrant, we take the angle as π - θ as we want to identify the coefficients in $-x$ and y axes. In the third, we calculate the angle as θ - π to identify in $-x$ and $-y$ axes. Finally, in the fourth quadrant, we calculate the angle as 2*π - θ.

5.3.3 Aggregated Vector

A simple overlay of all the vectors creates a cluttered visualization, and the vectors often depict inconsistent directions due to the wiggle in the isoline. Hence it is hard to understand the global pattern (if any). Therefore, we next aggregated the vectors, as follows.

Let $\vec{V_X}(t_i)$ be a vector representing an increasing trend for the datapoint X at a timestamp t_i. After the

shifting and subtraction, we get a matrix of $M \times N$ vectors in 24 directions for X in a timestamp. For each data point, we aggregate all these these 24 vectors to compute $\vec{V_X}(t_i)$.

$$\vec{V_X}(t_i) = \sum_{i=0}^{23} \vec{V_X}(P_i) \times I(P_i), \tag{5.4}$$

where $I(P_i)$ is an indicator variable which is set to 1 for the increasing trend, and 0 otherwise. Similarly, we also compute the decreasing trends.

We compute the aggregated vector $\vec{V_X}$ by aggregating all the 2D vector components for all timestamps t_1, t_2,...,t_n.

$$\vec{V_X} = \sum_{i=1}^{n} \vec{V_X}(t_i). \tag{5.5}$$

5.3.4 Final Vector

Let u_X be the vertex in the weighted graph G that corresponds to the location X. The vector V_X represents the differential trends acting along its adjacent edges. Therefore, while calculating the final vector, we scale the magnitude of the vector by the average weight w of the edges adjacent to u_X.

We compute the final vector from the aggregated vectors and the previously computed weighted graph. To overlay the simplified contour map with resultant vectors computed in the previous step, we aggregate the vector components of each point on a path into a single vector. Previously when we calculated vectors, a segment with different data points had their own individual vector data. This creates crossing of the directions on the same segment. To better depict the trend, for each simplified contour segment (p_0, p_k), we aggregate the vectors associated to the path $p_1, \ldots, p_k$ that it simplified (Figure 5.3). Using the normalized weight calculated for any two points p_{i-1} and p_i, in the weighted graph, the final vector $\vec{V}$ is formulated as follows:

$$\vec{V} = \sum_{i=1}^{k} \frac{|p_{i-1}p_i| \cdot (\mathbf{v}(\mathbf{p_{i-1}}) + \mathbf{v}(\mathbf{p_i}))}{|p_{i-1}p_i|}, \tag{5.6}$$

where $\mathbf{v}(\mathbf{a})$ denotes the vector associated to the location a.

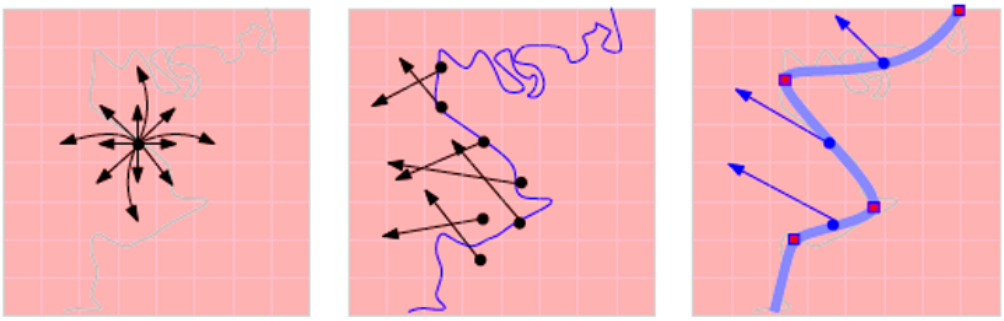

Figure 5.3: (left) Computing the vectors for a spatial location s. (middle) Illustration for vector clutter for a general contour line. (right) Aggregated vector overlay along a simplified contour line.

We also calculate the magnitude of this vector from its coefficients $\vec{V_x}$ and $\vec{V_y}$ from the following:

$$mag(\vec{V}) = \sqrt{\vec{V_x}^2 + \vec{V_y}^2} \tag{5.7}$$

We overlay the final vectors on top of the generated contour map. A comparison has been shown among the state of final visualization with and without this aggregation in Figure 5.3 and Figure 5.4. We use the magnitude value to identify the length of each vector in the final image.

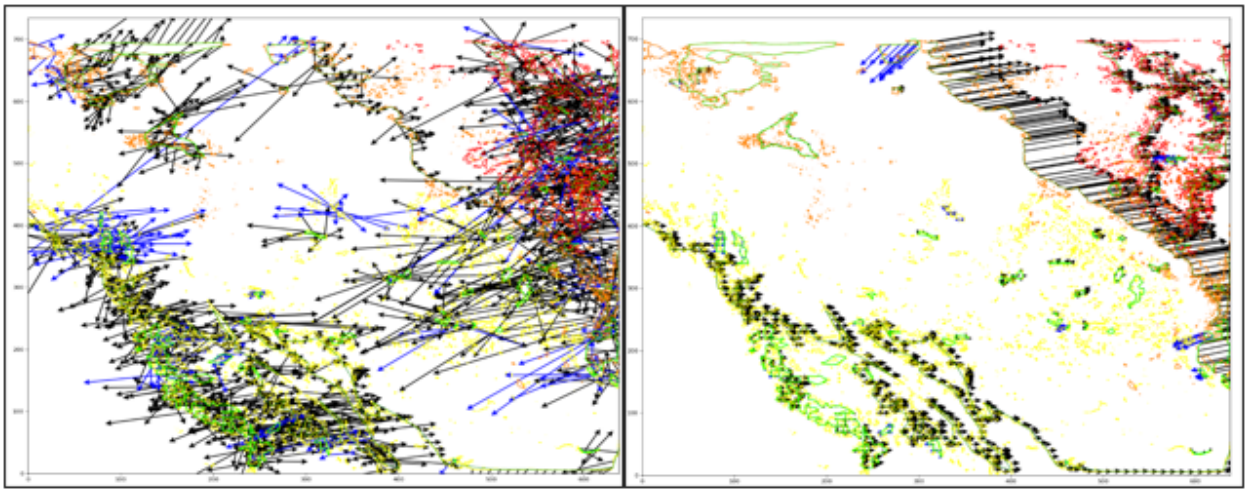

Figure 5.4: Illustration of contour map (left) without vector aggregation and (right) with vector aggregation.

5.4 Vector Rendering and Quadtree

Storing and processing large amount of data creates high computational complexity. Efficiently finding answers regarding spatial attributes require the use of Hierarchical data structure [124]. Adaptive representation of the spatial information [30] has been proposed in literature that replaces pixels with particles positioned according to image content on large amount of microscopic image data. The approach reduces both computational cost and memory cost of image-processing tasks where they represent the pixel as a graph and binary pruned tree.

To render the vectors interactively, we represent the 2D spatial location with a *quadtree* [124], a hierarchical data structure often used to represent geospatial data, where the space is recursively divided into 4 regions. We implement a point-region (PR) quadtree [123], which is a variable resolution representation of a data field in two dimensions. Each quadtree node is associated with a rectangular region of the $M \times N$ matrix, which represents the whole geographical location. Any geospatial variable regarding that rectangular location was stored at the corresponding quadtree node, with each node aggregating the information stored at its children nodes (e.g., see Figure 5.5).

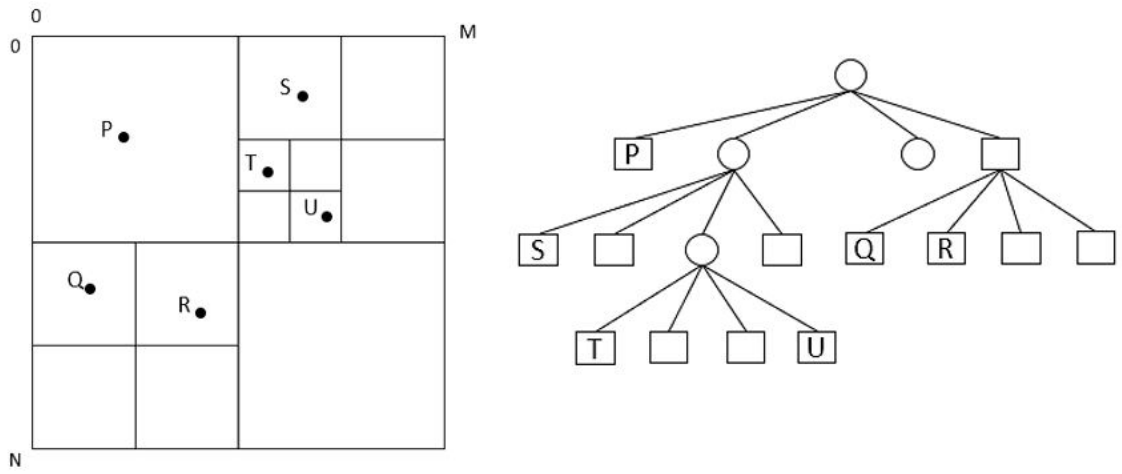

Figure 5.5: Illustration for a recursive spatial subdivision using a quadtree.

The algorithm for building spatial dimension in this process starts from (0, 0) and takes M as width and N as height. We use variation of quadtree depth to identify their significance in computational complexity and resolution in building the final visualization. We further explain our evaluation in Section 7. The construction is a bottom-up approach where four (x, y) pixels at the depth $N + 1$ are combined in one pixel at depth N. We collect information regarding each pixel starting from quadtree depth M, break the texture of resolution and then go to depth $M - 1$, where one quadrant represents four quadrant at depth M. A region $X \times Y$ with zero or one data point is represented by a PR quadtree consisting of a single leaf node. Any region containing more than one data point is split into four equal quadrants and each quadrant into four subsequent sub-quadrants until each leaf of the quadtree contains a maximum single point. To find a point p in the tree, beginning at the root, we continuously travel to quadrant that contains that point until search results in a leaf node. Since four different sub-quadrants contain different location values in x and y axis of the region, indexing of each branch node makes it easier for us to find the required result.

The queries in ContourDiff are analyzed with the region matching with the process of resolution or depth of decomposition. A region is divided until a count (threshold) of data objects per space structure is reached. We choose this procedure for two advantages: Firstly, we gather all information of a geospatial variable across the landscape for all nodes. ContourDiff works with dataset over a large time span and computation as well as collection of information of a node with respect to 2-hop neighbors over n timestamps can be computationally expensive. Secondly, the process of interaction with the system and analyzing any query become faster. This procedure first uses the whole region of interest as its root. Then it searches for a higher number of nodes than the threshold provided. If it is successful, the space is recursively divided in four children region. Thus, a high threshold or depth leads to a decrease in number of nodes from the weighted graph. When the nodes and consequently paths over the map is reduced, a decrease in number of vectors is observed in a region.

We allow users to choose the quadtree resolution, at a low resolution, the recursive division of space continues until it hits the given resolution. The vectors are plotted at the intersection points of the quadtree

with the contour lines. Using this makes the vector rending interactive (a few seconds) and even within milliseconds for a higher resolution.

5.5 Pattern Validation using Synthetic Data

This section explains how we validate our methodology and read the generated vectors on top of the contour map. We explain the ContourDiff visualization using synthetic data. This will at first, familiarize the users with the basic differential trend patterns and how to read them. Secondly, this also helps validate the system functionality since the knowledge of the extracted features and the chance to modify the experimental conditions allow us to draw several insight to build our visual design later on. While the generated synthetic data is used as a test case of evaluating our methodology, we understand that the real world dataset pose more complex challenges. The synthetic dataset is a means of exploring the design choices of different steps of our data analysis and visualization workflow.

5.5.1 Objectives

Synthetic data can be described as the data that is generated by the simulated users for in a simulated system, performing simulated actions. The primary objective of designing a synthetic dataset of our own to understand two scenarios - A) How does a region of interest with different contour lines can be combined with vectors? We investigate this by creating a synthetic matrix dataset where we design a contour map with different contour lines and later vectors are overlapped along the contour lines to show changes from one part of the region to another, differentiated by the lines themselves. B) How can change of numerical value over different timestamps impact the visualization?

We create two dataset of same size with different values and then we combine the generated vectors from each timestamp to overlay them on top of one of the contour maps. We explain the generation of these dataset in Chapter 3. We also explore the scalability issue compared to a large scale geospatial dataset and how we can overcome the challenges associated to it. We do not change any step of the methodology for evaluation so that the changes through the workflow become constant for our synthetic dataset and large-scale geospatial dataset.

5.5.2 Interpretation

We generate two timestamps, as illustrated in Figure 5.6A and Figure 5.6B. The corresponding vectors at each timestamps are depicted in Figure 5.6C and Figure 5.6D, respectively. One can immediately see from the vector sizes that the rate of change along the contour lines is much higher at timestamp-B than that of timestamp-A. Finally, the increasing (black) and decreasing (blue) vectors are aggregated and plotted on the contour map in Figure 5.6A. One can see from Figure 5.6E that the vectors indicated by black arrows show the increasing (low to high) changes along the contour lines. The black arrows depict the opposite trend.

While we identify that the real life complex dataset does not create a simple concentric contour lines, this approach is vital in understanding the increasing and decreasing data distribution.

5.6 Summary

We have discussed the strategy of trend vector computation as a representation of change in this chapter. We have provided an in-depth and detailed step-by-step computation of increasing and decreasing vectors from a non-directional dataset. Furthermore, we have designed the ContourDiff visualization with a contourmap based on the data distribution. We have employed a data structure, Quadtree, to efficiently store and process large amount of data. Finally, we have built a synthetic dataset and generate ContourDiff images with them to provide overall readability of the differential patterns. Not only the design choices in our workflow have been explored with the dataset, but also we gain intuitive understanding about how ContourDiff can work with real-life datasets.

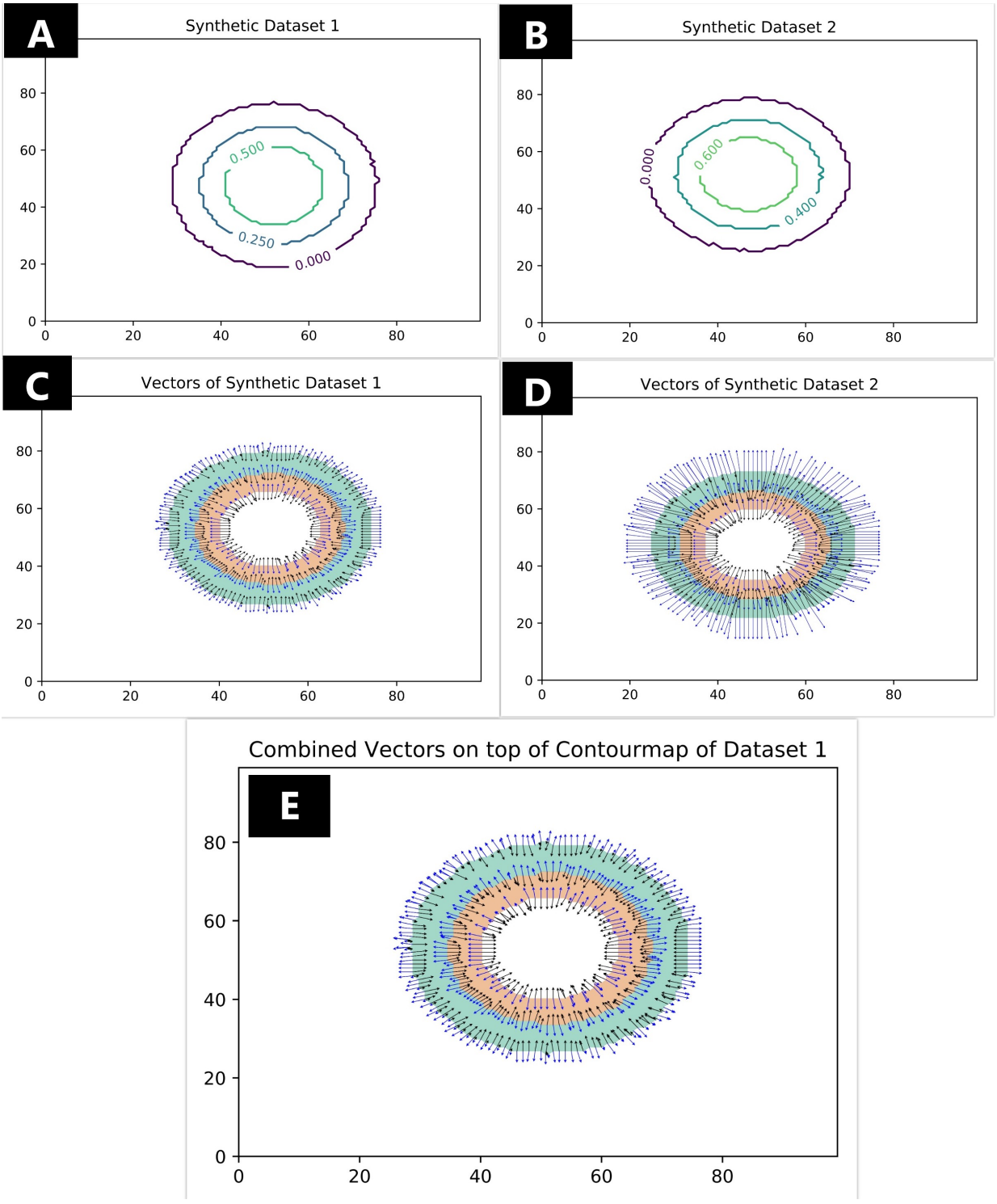

Figure 5.6: Illustration for the ContourDiff visualization on a synthetic dataset. (A) Contourmap of synthetic dataset 1; (B) Contourmap of synthetic dataset 2; (C) Vectors generated from synthetic dataset 1 on top of its own contourmap; (D) Vectors generated from synthetic dataset 2 on top of its own contourmap; (E) Combined vectors generated from synthetic dataset 1 and 2 on top of the contourmap generated from dataset 1.

6 Visual Design

In this section, we explain the detailed support to execute tasks by the users interacting with the system. We create a standard graphical interface by following the principal for grouping interactive controls with similar functionalities, to provide users with an intuitive workflow [144]. Parts with similar functionalities are designed closely and grouped together in the interface, as shown in Figure 6.1.

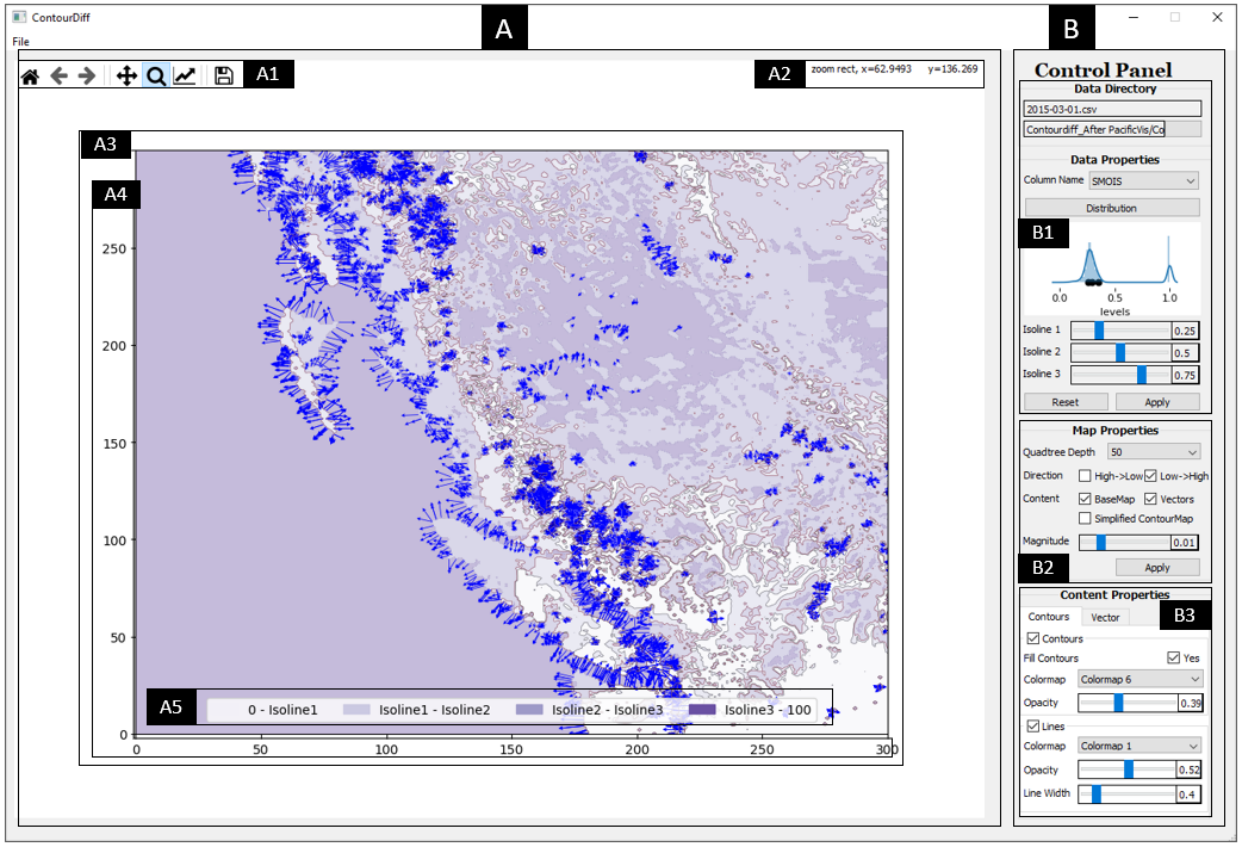

Figure 6.1: Graphical User Interface design of ContourDiff.

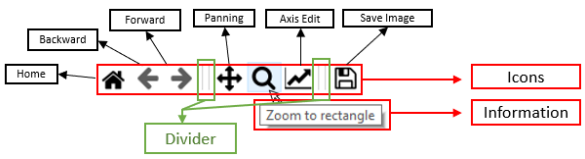

Figure 6.2: Icons and their corresponding information in ContourDiff.

ContourDiff is composed of an *Evolution View Panel* on the left side (Figure 6.1A) and a *Control Panel* on the right side (Figure 6.1B).

The *Evolution View Panel* is divided in five components as illustrated in Figure 6.1. The main display portion where the created visualization is presented is shown in Figure 6.1A3 and the direct interaction techniques are provided in Figure 6.1A1. Figure 6.1A4 and Figure 6.1A5 are two sub-components of the main display that indicates $x \times y$ portion showed on the screen and the colormap chosen to display the contourmap respectively. Figure 6.1A2 indicates the current location of the cursor and whether the entire display is in zoom/pan state or not. In the evolution view panel, we provide primarily a basic contour map design based on the user defined data characteristics. The dynamic change of geospatial variable is depicted by a pattern flow on top of the contour map. The required context for analyzing temporal changes can be gathered by the location of a certain region and its latest timestamp contour map.

The *Control Panel* is mainly divided into three subsections. Figure 6.1B1 allows users to explore data characterises of the latest timestamp of the selected dataset and choose isolines to create the contourmap. Figure 6.1B2 helps users to change data mapping and Figure 6.1A3 helps to edit visual properties of the contoumap and vectors. By selecting a plethora of conditions available from the control panel, the users can highlight a region and examine the detailed process of change scenario. We also provide a number of interactive options to make the visualization interesting and more readable to users. A detailed description of the system interface and their intended tasks are as follows.

6.1 Evolution View panel

Evolution view panel is mainly designed for the display of ContourDiff visualization and direct interaction for rendering vector patterns, explore the trends of vectors and collect the patterns by saving the current visualization in an image.

6.1.1 Basemap Design

To encode spatial information regarding a single geospatial variable, we design a basic map of the region of interest. Different isolines of contour map based on data distribution of the most recent timestamp is created in an image wrapper. The contour map is then colored with a perceptual color map to create segmentation of the region based on the distribution. We place a legend at the bottom and middle position of the image wrapper to easily identify the differentiation. Based on data pattern, the image wrapper is tagged with location point on its x and y axis. Hence, the users can select a particular region of the map to investigate a region and follow their distribution. The users can also familiarize themselves with the land pattern from a geographical map and then identify similar region of interest based on data patterns. Additional statistics of data distribution will be discussed later in control panel.

6.1.2 Rendering Vector Patterns

We employ a pattern-based visualization of user-selected timeline on top of the colored contour map. The patterns are designed as vector-based glyphs to identify various levels of change scenarios. The contour lines have been used as the base of the vectors, so that the users can easily identify the change of the geospatial data of a region. Primarily two types of vectors are visualized to identify the pattern over the whole area. The aggregated vector over a single data point has been categorized as increasing, low change to high and decreasing, high change to low.

For the vector overlay, we design vectors as 'free-flow', i.e., vectors can be presented from 0 to 360 degree angle. We also encode the change magnitude as the length of the vector for two purposes: (a) understanding the differential trend at a particular spatial location by locating prominent trends along a contour line, and (b) comparing the trend among several similar regions of interest (e.g. water bodies for soil moisture). Instead of using a colormap, we use basic colors to portray these vectors so that they can be easily identifiable. We design the vectors with same thickness to reduce any visual clutter and enhance readability for users.

6.1.3 Trend Exploration

Since the primary visualization provides an overview of the whole geographical area, it is sometimes difficult for readers to understand the trend at a finer spatial resolution. Therefore, we provide several direct interaction features such as zoom, pan and image-save capabilities, as illustrated in Figure 6.1A1. They can also choose a quadtree parameter to obtain a sparse overlay of the vectors. We employ these interactions as a horizontal ribbon-like panel at the top-left corner of the panel. This panel contains icons which, while hovered, presents clue about their functionality as shown in Figure 6.2. We use icon-label dual tactics for two reasons: firstly, perceptions of ease of use and usefulness is better than label-only interface in an end user application [156] and secondly, it provides easy implementation of further options in the future.

Now, given a relatively large number of vectors in the whole geographical area, it would be very difficult for users to discern them with minimal changes as they look similar, let alone distinguish from each other. To improve this scenario and for an in-depth analysis of change pattern, we allow users to zoom in by selecting a region of interest and a fine-grained visualization of the corresponding region will be shown over the whole image wrapper (Figure 6.1A3). The axes and location settings will be automatically updated to indicate the changes. The user then can hover the region enabling the panning feature.

All the changes made by the user in terms of zooming and panning are saved in the system such that the user can navigate through these changes using the *Forward* and *Backward* features from the interaction panel.

6.1.4 Pattern Collection

The change patterns designed by users may provide a certain understanding of a region for their own set of tasks. While an animation or side-by-side analysis of geospatial data over time can provide an understanding of the region, we provide similar understanding in one image frame. Hence, we provide a pattern collection feature, by which the users can save the final output of their design in their own device in different file formats. Before creating the image, the users can create title for the image and change the functionalities (Left axis point, Right axis point, label, scale) of x- and y-axes using feature from the Interaction Panel. These images can be used for further exploration and gain insight.

6.2 Control Panel

The Control Panel has been provided to explore the ContourDiff visualization presented in Evolution View Panel. We organize the panel according to the task process and provide three different modes of control to facilitate the exploration, namely, the data mode, to provide summary of the recent timestamp, the map mode, to provide control over the settings of the visualization, and the content mode, to enable users with various interaction possibilities for the created pattern.

6.2.1 Data Mode

Data mode controls how users identify input data and its characteristics in the control panel. This consists of following two sections:

A) **Data Directory**, which indicates the most recent timestamp for which original basemap is created as well as the historical dataset containing the other timestamps.The user can choose the dataset from any storage location on the local computer clicking the button on the right side of the *"observation directory..."*. Then the system sorts the files based on their timestamp information and selects most recent timestamp for the background map generation. This also shows the file name and observation directory information in *"Current file..."* and *"observation directory..."* locations respectively. This process is shown in Figure 6.3.

B) **Data Properties**, which visualizes the distribution of the most recent dataset to allow users choose any three contour thresholds and divide data into four intervals based on those thresholds to generate the contour map.

Based on the choice of an user from a number of geospatial variables, the system shows the distribution for that variable using a histogram plot. A default choice for creating the contour lines was set according to the lower quartile, median and higher quartile (i.e., the sliders for choosing quantiles were set at 0.25, 0.5 and 0.75, respectively). The distribution is shown as a histogram plot. The users use the slider controls to choose a set of contour thresholds (or, choose the default, i.e., quartiles) to generate the base contour plot. Inspection of the histogram also allows users to choose a different set of contour thresholds to generate

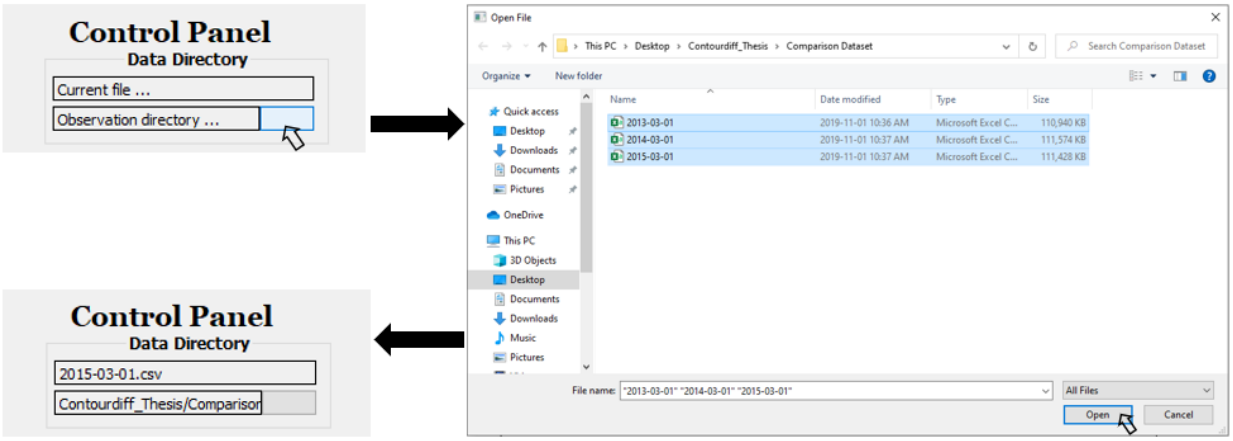

Figure 6.3: Depiction of how users can input dataset in the system.

required base contour plot. This is shown in Figure 6.4. Moreover, Based on data from other timestamps, resultant vector is then generated automatically by the system for visualization.

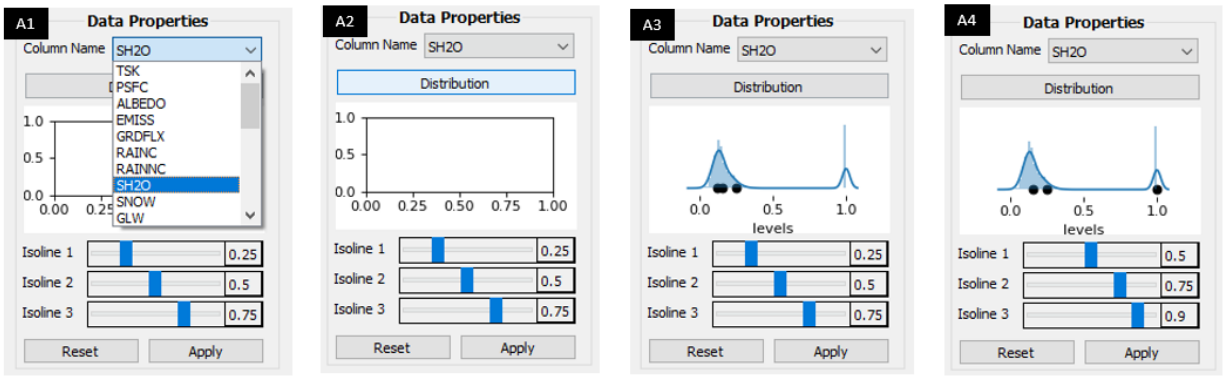

Figure 6.4: A1 shows different geospatial variables in WRF dataset, A2 shows the distribution wrapper, A3 shows automatic distribution of chosen variable and A4 shows data distribution with different isoline.

6.2.2 Map Mode

Map mode provides several options for users to change the characteristics of the image before final visualization. This includes the choice of quadtree depth for sparser vector layout and faster interaction. We previously discussed how ContourDiff uses quadtree to store information of all the (x, y) positions over the region. Choosing a value from the drop-down menu in a range of 10 to 100 allows users to control the density of the vector overlay and thus the speed interactive exploration.

Since the set of vectors are categorized based on the differential trends (increasing and decreasing), we also add direction controls to allow users to select the type of direction they want to show along the contour lines. Default is set to show both types of directions.

The final visualization has two components as previously discussed, contour map and vectors. The users

have the option to choose to visualize any one or both with respect to their interests. The default setting is to show vectors with color filled contour map. If the user chooses only vectors to visualize, the contour map will not be filled. Moreover, the magnitude control allows users to filter the vectors with low magnitude to reduce visual clutter. Figure 6.5 shows different configurations of map mode and their respective visualization.

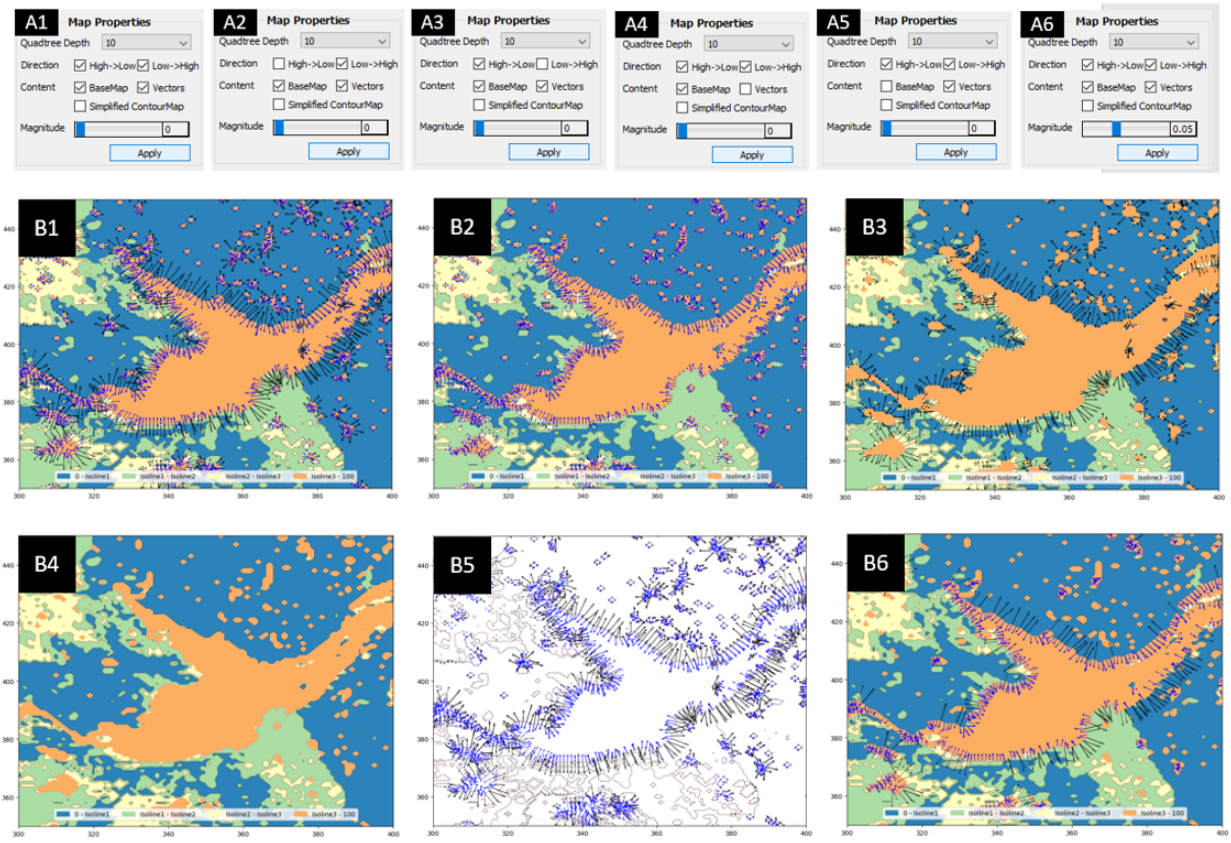

Figure 6.5: A1 to A6 show different configurations of map mode and B1 to B6 shows respective visualizations of that configuration for SH2O parameter from WRF dataset from March, 2013 to March, 2015.

6.2.3 Content Mode

The final image is created with a default colormap for both isolines and vectors. Moreover, vectors drawn are not primarily scaled to any range. We provide a number of interaction possibilities for both these cases such as choosing colormaps [58] for filling contour intervals, changing the color and thickness of the vectors, and scaling the vectors using different types of data mappings (e.g., logarithmic, linear, exponential).

Contourmap properties

The colormap indicates the color of four different classes of data in the generated contour based on chosen contour thresholds. Though Rainbow color map is used to visualize various geospatial information such

as temperature and humidity, it can confuse viewers through its lack of perceptual ordering and provides misleading information by obscuring data [25]. We generate a number of colormaps from Colorbrewer [58] to provide users with color scheme options for the contour plots, as shown in Figure 6.6. We show the created visualizations for these colormaps in Figure 6.7. As the color intensity can sometimes be distracting and overpowering for the visualization, the users can increase or decrease transparency from the opacity slider.

The contour lines can also be colored by choosing custom color, but the default is set to colormap of the contour intervals. A line width option is added to increases or decreases the thickness of the contour lines in the map.

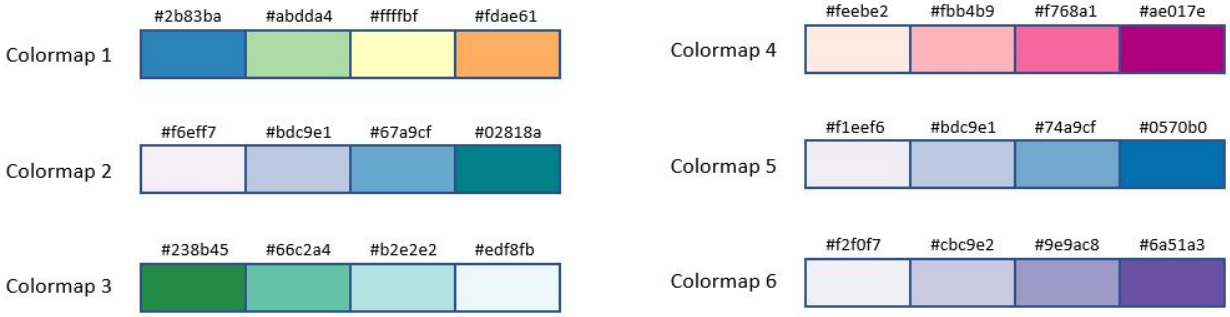

Figure 6.6: Colormaps and corresponding hexadecimal values for contours.

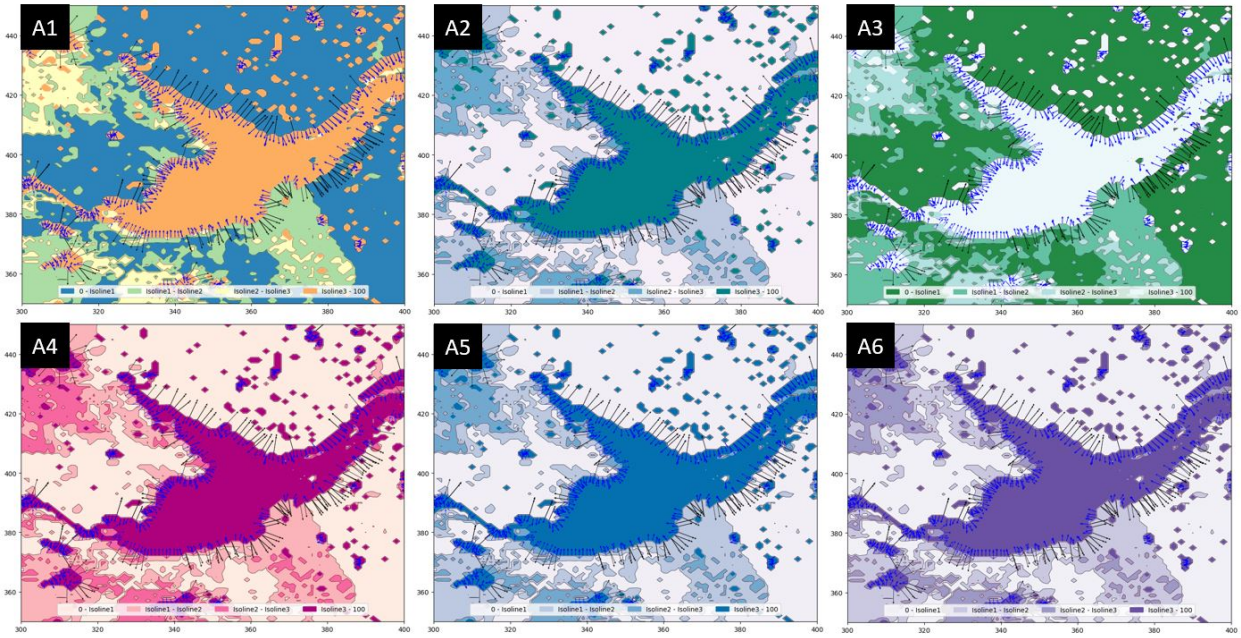

Figure 6.7: Visualization of SH2O variable from WRF dataset with different colormap for contour segments.

Vector map properties

The vectors are shown using arrows, and they have two properties — arrow scale and color. The users can change the arrow scale based on their preferences. There are four arrow scales implemented in the system. The linear arrow scale is the default that portrays the arrows as it is. The exponential and logarithmic scales changes arrows by changing the corresponding data mapping. Choosing normalized scale first normalizes all the magnitude values for all the vectors from 0 to 1 and then changes the arrows accordingly. The arrows are colored primarily with blue and black respectively for increasing trend (Low to High value) and decreasing trend (High to Low value) over time, respectively. But users may choose to use different colors using a drop down menu.

7 Performance Analysis

In this chapter, we discuss the performance details of our ContourDiff system and evaluation of our system generated visualizations. We at first provide an overall detail of machine specifications and libraries used to implement ContourDiff in 7.1. We also analyze the construction time of the visualization in different steps, examine the scalability of the system, and show the vector overlays by varying the leaf size (resolution) in this section. We then examine real-life scenarios using both numerical and image dataset and present our findings in section 7.2. We present a thorough expert evaluation of our system generated geological trends in section 7.3 to refer its usability in spatio-temporal phenomena. Finally, we discuss our overall findings in section 7.4 in relation to the research questions described that we previously describe in Chapter 1. The code for ContourDiff can be found in the following link: https://github.com/zar92/ContourDiff_Thesis

7.1 Implementation Details

7.1.1 Machine and Library details

The machine used to develop and evaluate ContourDiff has an Intel(R) Core(TM) i7-8700 Processor with 3.2 GHz processing speed and 64 bit operating system. The machine has 16 GB RAM and 1920×1080 display resolution. Dataset is stored locally in the machine outside the system. The system is designed as a desktop based application, where the main algorithm to build the visualization has been implemented in Python. The user interface components have been developed as QT widgets using PyQt5. We use quiver plot to draw the vectors over contour paths. The mechanism takes vector components (u,v) in X and Y directions to plot a vector at points (x,y). For user convenience, we use an uniform length and width of the arrows for the representation of vectors. To implement the ContourDiff approach, we used the libraries shown in Table 7.1.

The dataset used for this work has the size 639 × 699 with a resolution of $4km$, which is also the size of the underlying topological map. We report the results based on five different variables: Surface Skin Temperature (TSK), Planetary Boundary Layer Height (PBLH), Soil Liquid Water (SH2O), Ground Long Wave (GLW), Soil Moisture (SMOIS) from WRF dataset, each associated with a geospatial characteristics. We provide a sample data characteristics in Figure 7.1.

Name	Version	Description
Pandas	0.23.4	creating the initial and weighted graph from contour map
Numpy	1.15.4	Vector computation
Matplotlib	3.0.2	Creating interactive main canvas, distribution canvas menubar and plotting vectors
PyQt5	5.13.0	Designing the Graphical User Interface
Scikit Image	0.16.2	This module is used to read as well as processes RGB images into greyscale and transform them into matrices.
Time	—	Calculating time complexity of different modules of the system
Seaborn	0.9.0	Displaying data distribution

Table 7.1: Libraries used for implementation.

7.1.2 Construction Time

Construction time was the most critical issue while creating the ContourDiff visualization. The algorithm to construct the final image has been optimized through a series of modifications, which reduced both the processing time and interaction time.

First, the initial implementation iterated over dataframes while modeling the weighted graph, and thus creation of the graph took several minutes for a few timestamps as we created a dataframe by iterating over point locations on the contour path. Since such a dataframe contains paths in each level of the map, iterating over each point takes time. We modified it by creating a list of the dataframes and updating each column independently as a list of values. For three files, this modification reduces the time for modeling the graph significantly from an average of 93 seconds to average of 0.2 second.

Second, plotting the arrows using arrow location, magnitude and direction values takes significant time. This hinders making any changes to already generated image such as contour colormap, vector colormap, magnitude change. Initially, we provide these arguments as arrays of values. So, iterating over these values for each point in spatial dimension takes long time to visualize. Therefore, we updated the node values when we generate the weighted graph and simplify the contour map using quadtree. Before this modification, this step used to take more than two minutes for a simple three file visualization. But later on it takes an average of 0.02 second to update the changes.

Third, creation of weighted graph took an average of 25 seconds for three timestamps. Primarily we

Statistics/ Variable	TSK	PBLH	SH2O	GLW	SMOIS
mean	(265.55, 256.09, 260.08)	(303.34, 328.84, 355.91)	(0.37, 0.34, 0.341)	(256.01, 175, 199.77)	(0.45, 0.45, 0.445)
median	(267.53, 252.46, 262.77)	(194.76, 299.1, 321.51)	(0.22, 0.18, 0.17)	(269.12, 156.95, 198.25)	(0.32, 0.32, 0.31)
standard deviation	(12.86, 13.58, 13.2)	(294.7, 225.4, 257.95)	(0.33, 0.34, 0.339)	(57.93, 46.33, 50.7)	(0.29, 0.28, 0.29)
maximum	(290.64, 292, 288.07)	(2236.4, 1548.65, 1792.55)	(1, 1, 1)	(381.96, 371.88, 339.6)	(1, 1, 1)
minimum	(221.85, 228.12, 220.48)	(12.3, 12.93, 12.5)	(0.02, 0.2, 0.02)	(120.1, 101.06, 109.75)	(0.03, 0.07, 0.04)

Figure 7.1: Time spent for each step of ContourDiff for WRF geospatial variables with three timestamps: March 2013, March 2014, March 2015.

iterated over all data points on the path to check whether the path has one single point or not to discard them from further calculation. But later, we change the dataframe consisting all contour paths that have more than one node and then iterate over only on those paths to do further calculation. This small change decreases the time to around 6 seconds.

We present the time needed to design different parts of the whole process for different variables in Figure 7.2. We consider different parameters from the dataset and we analyze the statistics to choose variables of varied characteristics. These characteristics along with time calculated for different steps is shown in the figure. Variation of data range does not impact the vector and contour creation. Overall time varies from 13.5 to 16 seconds for three timestamps from 2013 to 2015.

7.1.3 Scalability

We analyze the scalability of our system by taking different number of files for different timestamps. The time taken is proportional to the increase of memory and number of elements in a timestamp. We show a detailed evaluation of this in Figure 7.3 by analyzing upto 18 datasets, each containing 699$\times$ 639 instances for a variable. ContourDiff processes over 1 million data points for a single geospatial characteristic under a minute.

7.1.4 Quadtree Depth

The data processing and rendering time can be controlled by choosing different quadtree leaf size. We illustrate these impacts in Figure 7.4.

7.2 Case Studies

In this section, we discuss two case studies that provides qualitative analysis of our visualization approach and interactive user interface. We use Soil Moisture (SMOIS) and Soil Water (SH2O) from Weather Research and Forecasting dataset for one case study and Human movement image dataset to provide this evaluation

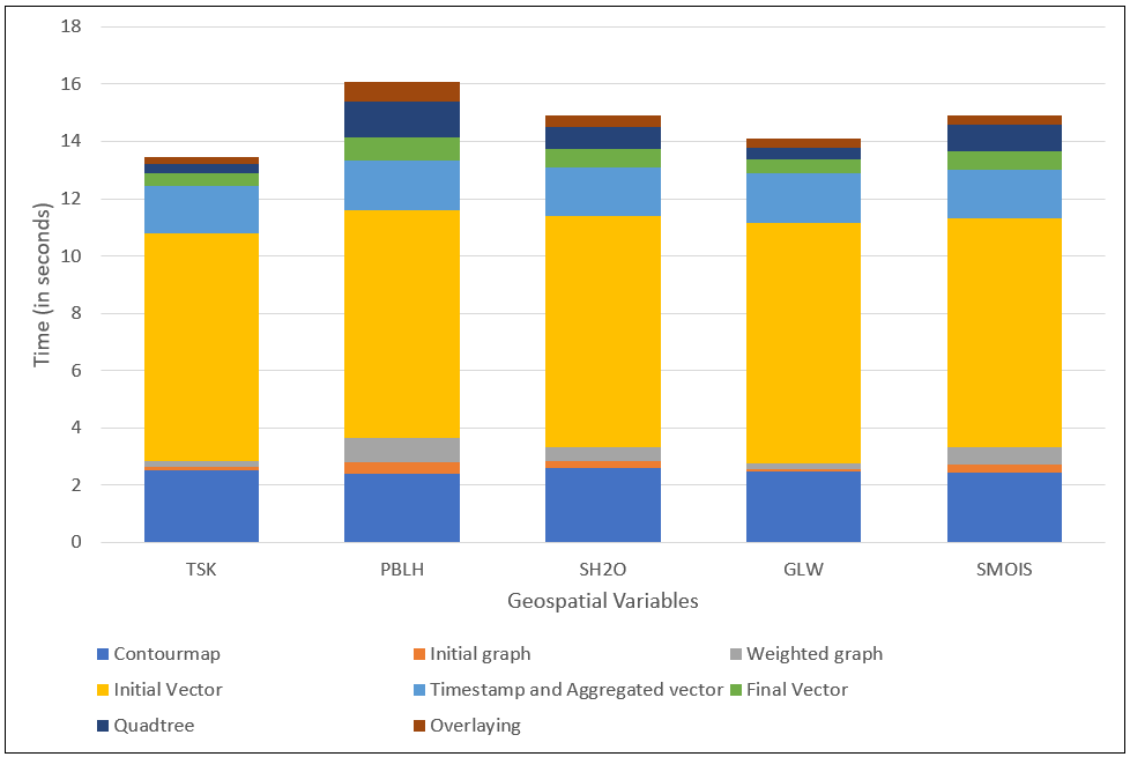

Figure 7.2: Time spent for each step of ContourDiff for WRF geospatial variables with three timestamps: March 2013, March 2014, March 2015.

of our system. These scenarios also prove our system is compatible with both numerical dataset (e.g., in csv format) and image dataset.

7.2.1 Case Study 1 (Geological Trends)

Image Generation: Here, we observe the patterns in *soil moisture* and *soil liquid water* content in Figure 7.5 and Figure 7.6, respectively. To generate both these images, we use March 2013, March 2014 and March 2015 dataset of WRF model output. We use same data distribution (0.15, 0.5, 0.75) to create contourmap from the March 2015 dataset. We also choose basic *Quadtree Depth = 10* and *Magnitude value = 0* to show all the generated vectors from our calculation. The colormap for contourmap and vectors in both these images are same as well. We choose *vector arrow scale = 'Linear'* for mapping vectors along the contour lines.

Findings: The soil moisture is saturated at the water body and the differential trend of soil moisture is generally affected by the attraction of soil surfaces to water [8]. Hence, we see the directions in the whole region mainly alongside the water bodies. For example, Figure 7.5A1 shows the differential trends along the contour lines over the whole Western Canada region. Figure 7.5A2 and Figure 7.5A3 are zoomed-in captures of the Great Bear Lake and Great Slave Lake, respectively. Both these images show a high soil moisture

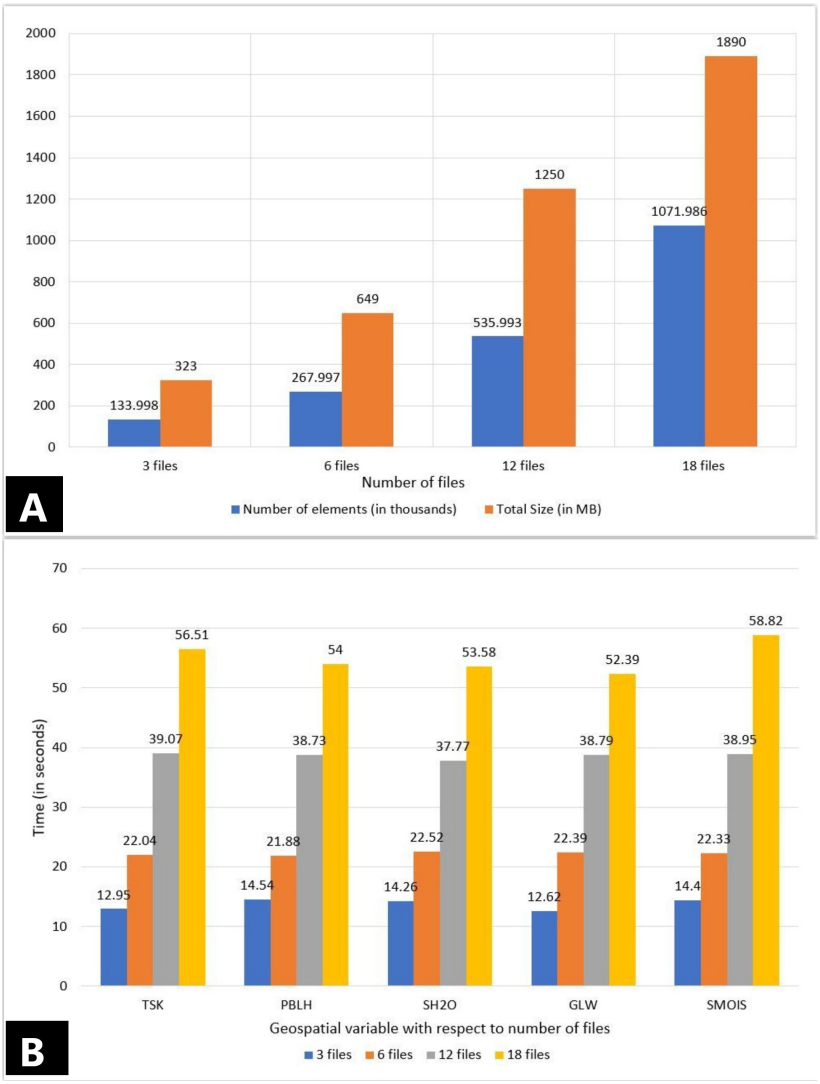

Figure 7.3: Scalability analysis for different number of timestamps in ContourDiff System - (A) Number of elements and total size in n number of files; (B) Total time needed for generating ContourDiff image from n number of files with 5 geospatial variables.

content in the lake region, but less around the lake in March 2015. Since the blue arrows and black arrows indicate low to high (increasing) trend and high to low (decreasing) trend over the map respectively, we can say from the vector size and direction, the soil moisture change over time along the contour line is decreasing.

Now, SH2O is only the liquid water content. For temperatures above freezing, the values will be the same. When the temperature is below freezing, at least part of the soil will be ice particles, meaning the value of SH2O will decrease. Figures 7.6A1, 7.6A2 and 7.6A3 show the change patterns over the whole Western Canada, the Great Bear lake and the Great Slave lake respectively.

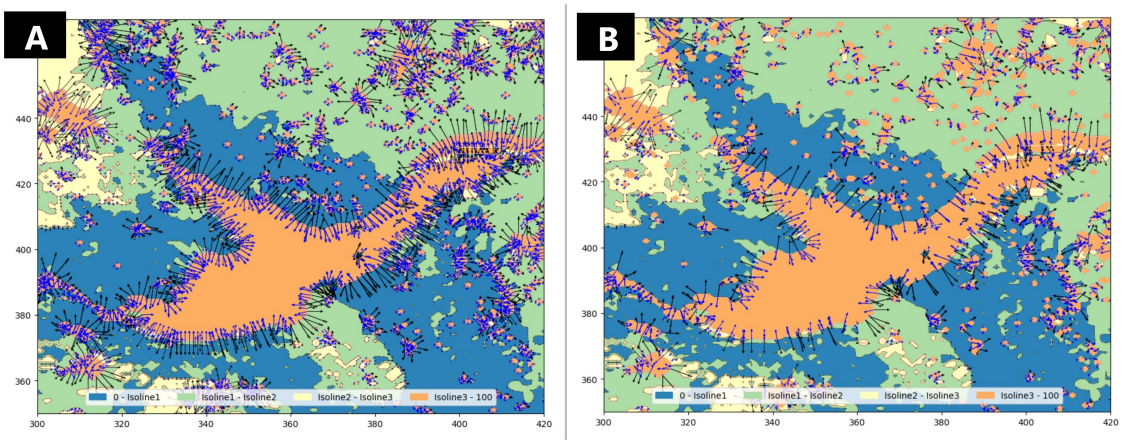

Figure 7.4: ContourDiff with (A) high and (B) low leaf resolution.

7.2.2 Case Study 2 (Motion Analysis)

ContourDiff is able to interpret small changes in image or numeric data. We show this in Figure 7.7 using the motion capture images of a human gradually standing up [91]. We primarily use images of Figure 7.7A1 and A2 to produce Figure 7.7B. Figure 7.7A3 shows the person approaching the final state. We see the increasing trends primarily on the top as the human is moving his head and shoulder upwards while changing his position. Although it is quite difficult for users to interpret detailed body movement from almost identical images, ContourDiff visualization could reveal both the significant change (head and shoulder) and an overall understanding of the movement (upward motion).

Such movement and the motion of the whole body is generated image with ContourDiff. This is advantageous for two reasons: (1) Understanding small yet significant movements in a system, (2) Analyzing the pattern of movement in different section of the whole region.

7.3 Expert Evaluation of Geological Trends

To examine the usefulness of the change patterns for scientific analysis, we evaluate them with two experts which are addressed in this section. The first expert (referred as E1) is a hydrology modelling analyst. The second expert (referred as E2) is a postdoctoral researcher working on ensemble hydrological forecasting at an University.

7.3.1 Evaluation Structure

The expert interviews were conducted separately through online video sessions. During both these interview, we first provided an introduction of ContourDiff and how they work. We generated images in front of them and explained how the contourmap was visualized from data distribution. Moreover, we discussed how two different vectors were computed and their significance in our overall trend visualization.

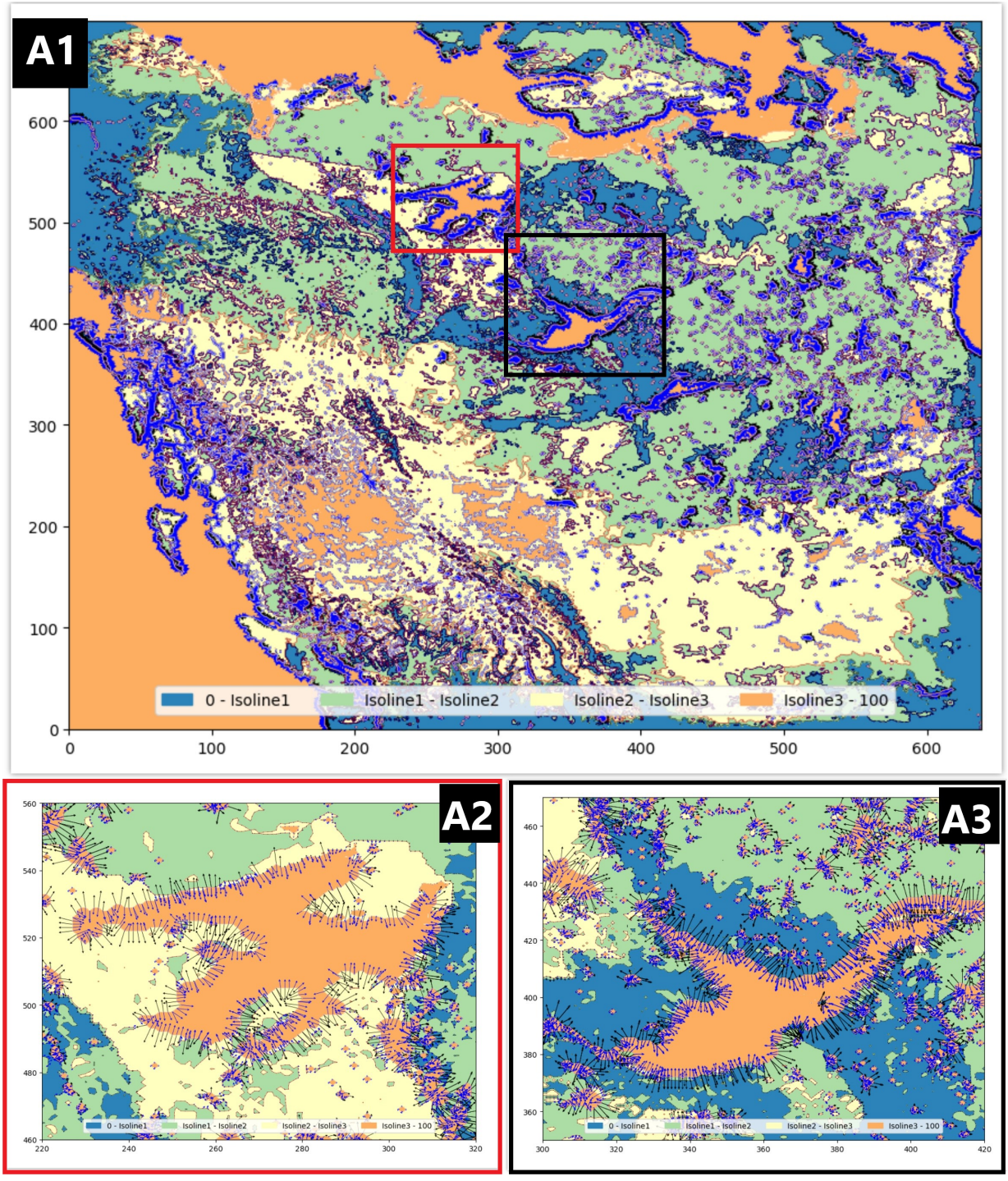

Figure 7.5: ContourDiff visualization for Soil Moisture (SMOIS) from March 2013 - March 2015 over (A1) Western Canada, (A2) the Great Bear lake, and (A3) the Great Slave lake.

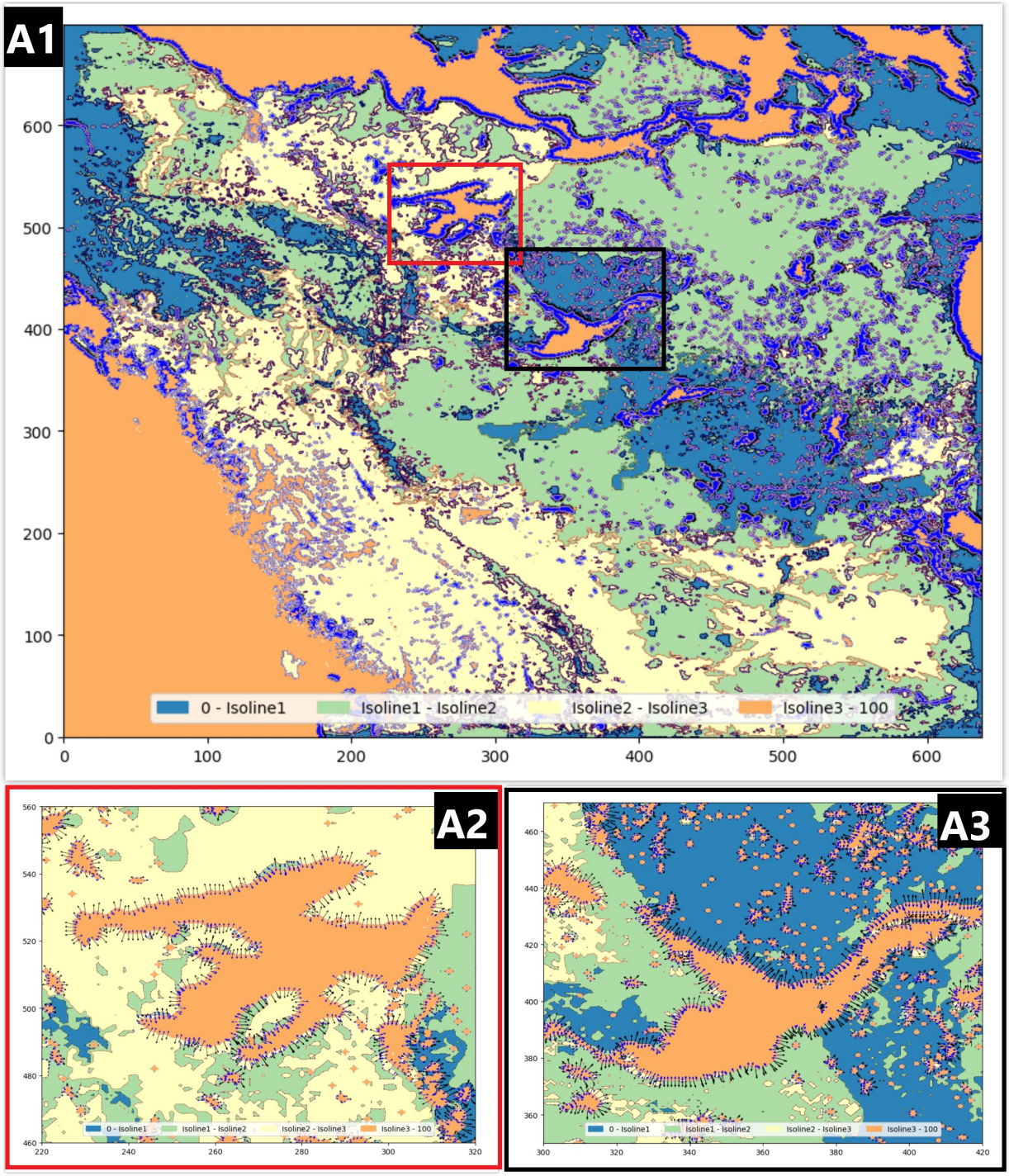

Figure 7.6: ContourDiff visualization for Soil Water (SH2O) from March 2013 - March 2015 over (A1) Western Canada, (A2) the Great Bear lake, and (A3) the Great Slave lake.

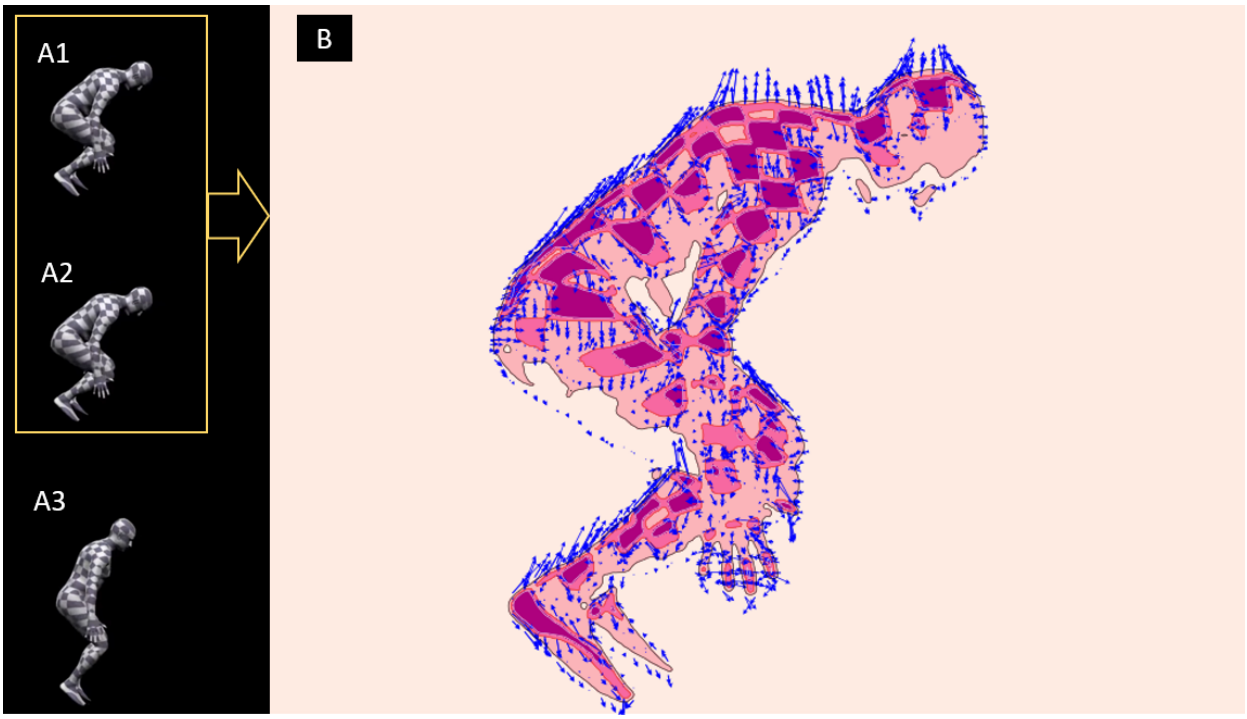

Figure 7.7: Motion analysis of a human gradually standing up. (B) is generated from the changes between (A1) and (A2); (A3) is a forward state.

After that, we showed the experts various images generated from ContourDiff using SMOIS and SH2O variables of WRF dataset. Though they had prior knowledge about the dataset, it was the first time for them to analyze the dataset from the perspective of trend patterns with vectors. The vector patterns along with contour plot was extensively discussed with both of them to understand the changes in regions with a large water body, especially, the Great Bear lake and the Great Slave lake. We have gathered key observations from their responses which can be utilized to assess the strength of our visualization framework. A summarization of the insights and the experts' comments is as follows.

7.3.2 E1 Interview and Evaluation

E1 interview was conducted on an initial development to gather insight about the correctness and usefulness of our visualization framework. We curated ContourDiff images with following considerations in mind.

- Initial assessment of ContourDiff and its' portrayal of contour gradients with *increasing* and *decreasing* vector trends over a geological region.
- Understanding potential impact of trend vector visualization with geological phenomena.

Image Generation

E1 was at first familiarized with the synthetic dataset and how ContourDiff generates visualization with them. Then we discussed the real life dataset. E1 was provided with Figures 7.5 and 7.6 to achieve aforementioned

goals.

Findings

The expert first looked into vectors over the whole Western Canada region from 7.5A1 and 7.6A1. Since the vectors in these images are difficult to read, E1 assessed the patterns over the Great Slave lake shown in 7.5A3 and 7.6A3. The findings are as follows.

- The soil moisture is saturated at the water body, which could be the reason that we see major differential trends (large vectors) mainly alongside the water body
- The decreasing trends (black arrows) are larger towards the land compared to the increasing trends (blue arrows) which are inside the water body. This indicates a sharp drop in soil moisture content as we move away from the water body. E1 also mentioned that such findings may sometimes give valuable information about the hydrological model that generated this soil moisture data, as well as creates the ability to ask new research questions.
- Since the soil moisture consists of soil ice and liquid water content, the soil moisture vectors appeared to be larger compared to that of soil liquid water. This was interesting since the expert could relate two different variables from two different ContourDiff visualizations.

7.3.3 E2 Interview and Evaluation

E2 interview was conducted on the fully developed system to analyze SMOIS images generated from ContourDiff and gain insight about the our overall visualization approach and its value in providing a summarization of long term changes. We curated various types of images to examine ContourDiff output against side-by-side contour plots.

We also provided an overall demonstration of how a ContourDiff visualization is generated using the tool. A detailed, step-by-step explanation of the framework was given to make E2 understand the general workflow of our approach. While creating the image, we discussed various interaction features as well. Since E2 has experience working with analyzing temporal variation of geospatial variable data and a number of software for weather forecasting, the introduction of our system was not time consuming. The interview was conducted over 2 days, each day with 1 hour of formal evaluation session. Moreover, since the expert is currently working on 2D plotting of snow water changes over time, abstraction of our research was not difficult. She commented *"Historical trend plotting is very important in hydrological research, especially with variables such as snow, soil moisture and precipitation."*

Task Description and Image Generation

We designed 3 set of tasks for E2. Each task was associated with three images, totaling 9 images for the evaluation. The tasks were designed to compare both short term and long term changes, and differentiate

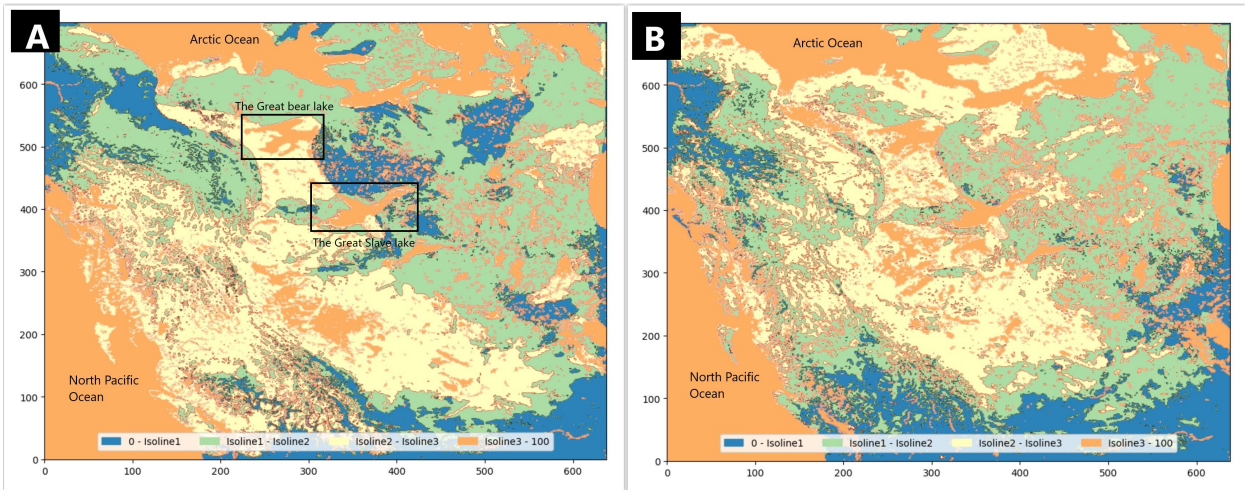

Figure 7.8: Soil Moisture Contourmap over Western Canada for (A) Winter 2013 and (B) Summer 2013.

between side-by-side and ContourDiff visualizations.

- **Task 1** *(Search and Locate)* - E2 was asked about the noticeable differences in soil moisture values over a short period of time. Images 7.8, 7.9, and 7.10 were prepared for this task with the dataset of Winter (February, March, April) 2013 and Summer (June, July, August) 2013. We requested E2 to locate the regions where soil moisture is higher or lower compared to its neighbouring areas from contourmaps (7.8) or trend vectors (7.9, 7.10).

- **Task 2** *(Identify and Compare)* - E2 was asked to interpret differences in soil moisture values over a longer period of time from side-by-side images and trend vectors. Images 7.11, 7.12, and 7.13 were prepared for this task with the dataset of March 2013, March 2014 and March 2015. We requested E2 to identify changes shown in individual contourmaps, and then compare them with the trend of change shown in ContourDiff regarding similarity and difficulty.

- **Task 3** *(Association)* - E2 was asked to differentiate soil moisture in Summer and Fall over a longer period of time. Images 7.14, 7.15, and 7.16 were prepared for this task with the dataset of Winter (March 2013, March 2014, March 2015) and Fall (September 2013, September 2014, September 2015). We requested E2 to provide us reasoning over these long term changes.

Findings

The findings of the aforementioned tasks are stated below.

- **Task 1** - From Figure 7.8, E2 located a number of prominent changes over the contourmaps. The expert decided to understand different colors and what they represent in the contourmap at first. Then E2 searched the regions alongside Arctic Ocean and North Pacific Ocean because she commented: *"Oceans*

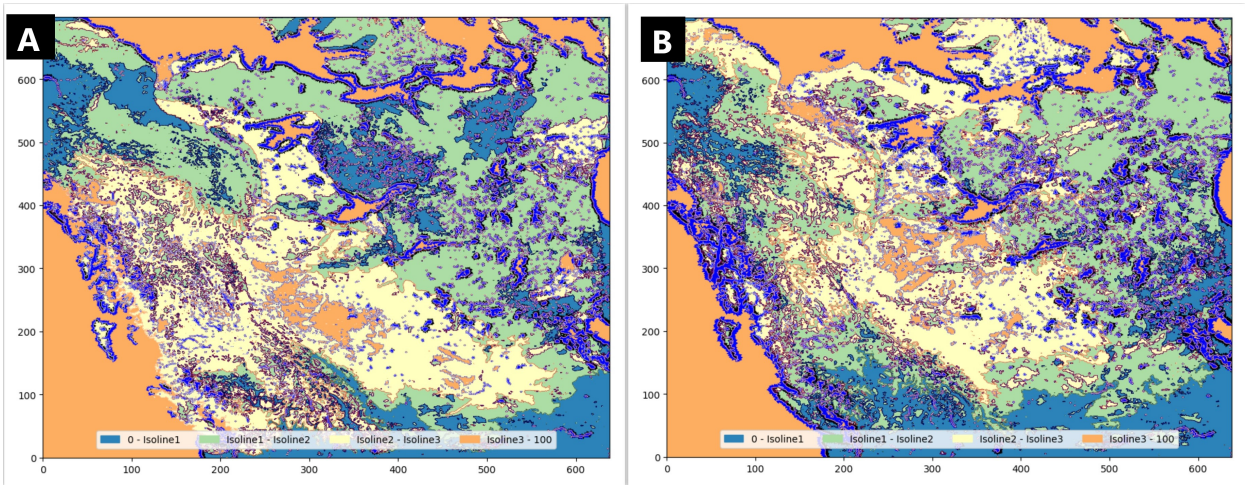

Figure 7.9: Trend vectors of Soil Moisture along the contours over Western Canada for (A) Winter 2013 and (B) Summer 2013.

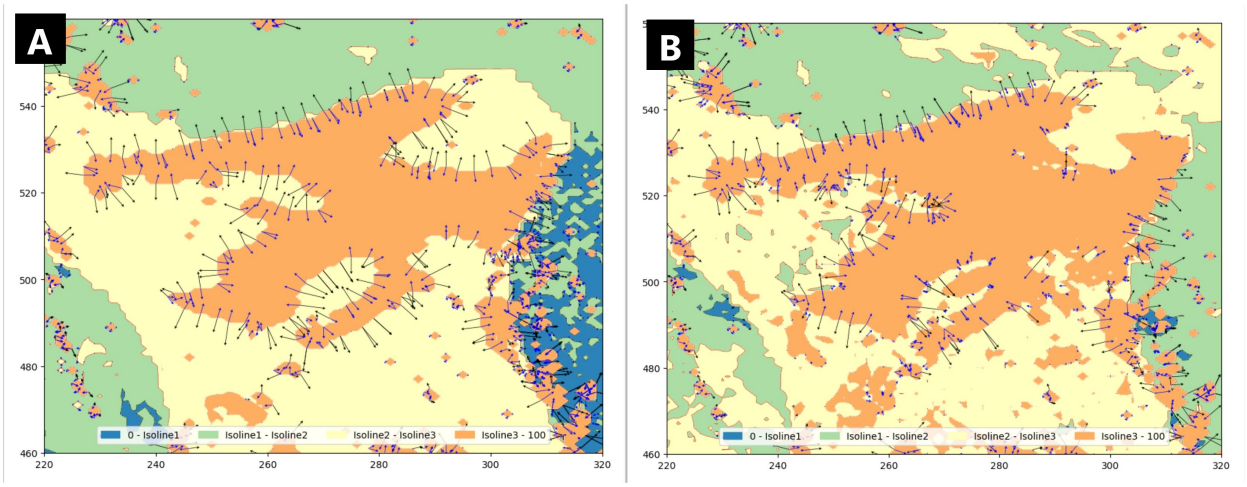

Figure 7.10: Trend vectors of Soil Moisture along the contours over the Great Bear lake for (A) Winter 2013 and (B) Summer 2013.

and lakes, and surrounding region have higher soil moisture value than land". She confirmed this by pointing out that in both Winter and Summer, the water bodies have significantly higher soil moisture content (colored with *orange* in contourmaps) than surrounding land regions. She also located several changes happening over this short period of time. For example, regions connected by Arctic Ocean on the top of the image has higher soil moisture in Summer than Winter. She also pointed out the bottom area of the image and North Pacific Ocean side regions where soil moisture is getting lower. She commented *"Canadian Rockies, and Southwestern Canada is drier in Summer than Winter, which makes a lot of sense"*. The expert indicated the a rise in soil moisture alongside the Great Bear lake. She pointed out that precipitation from water bodies cause higher soil moisture content in Summer which could cause that. The expert was interested to see trend vectors along the main water body regions in Figure 7.9. She identified the 'blue' vectors more prominent than the 'black' vectors along these regions as the soil moisture values are significantly higher in both Winter and Summer. Also, she identified the differences in number of vectors along the Ocean regions. She commented: *"Transition from ocean to dry land creates more gradient in Winter along the Arctic ocean"*. Moreover, she pointed out the elevation in North Pacific ocean shore that caused high soil moisture due to rains in Summer. However, looking at the Figure 7.10, the expert found short time changes not as interesting or easier to interpret than the contourmaps themselves. Because the changes over the Great Bear lake were almost insignificant except the rightmost region where the soil moisture was a little bit higher in Summer. She expressed that change can be easily understood from contourmap only and the vectors were very small probably due to the change being very small. We address this issue by expanding the timeline to 3 years in Task 2 and Task 3.

- **Task 2** - E2 was very interested in seeing a overall change in 3 years of data reflected by our ContourDiff visualization as she could compare them from the original contourmaps of that timeline. For example, Figure 7.11A shows side-by-side images of contourmaps for soil moisture obtained from March 2013, March 2014 and March 2015 dataset and the similar change is shown with trend vector plot in Figure 7.11B. The expert readily understood some big changes from side-by-side contourmaps but also expressed concerns over changes that are not quite so prominent. She addressed the issue by commenting: *"Over a large timeline, the changes are more interesting and difficult to interpret all changes at once, even with a smaller region"*. She correctly interpreted from 7.11B that the contourmap underneath is from the latest timestamp. Additionally, she pointed out the region around the Great Slave lake is getting drier because the soil moisture value is decreasing. She then checked for the trend vectors and found out 'black' arrows to be larger than 'blue' arrows and going from the lake to outside. She indicated that since the soil moisture value is decreasing on the outside over time, the trend vectors appear to be correct. Changes around the Great Bear lake is not so prominent as there is not much change in soil moisture over time, as shown in Figure 7.12. Even though she familiarized with two zoomed in images over the lakes, when we showed her the contourmaps and trend vectors over the

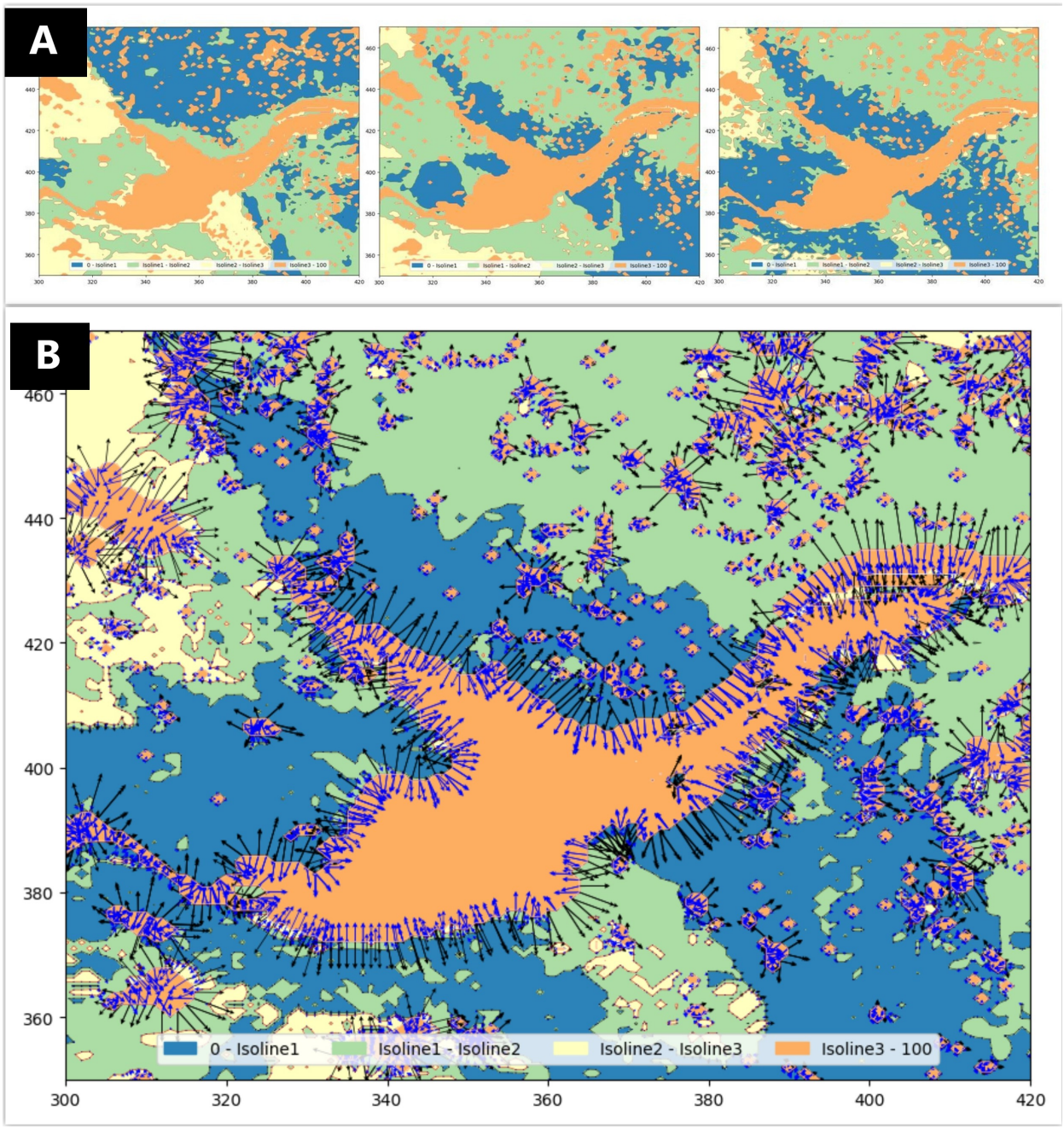

Figure 7.11: (A) Side-by-side contourmaps of Soil Moisture over the Great Slave lake for March 2013 (left), March 2014 (middle), and March 2015 (right); (B) Trend vectors from March 2013 to March 2015 of Soil Moisture along the contours of March 2015.

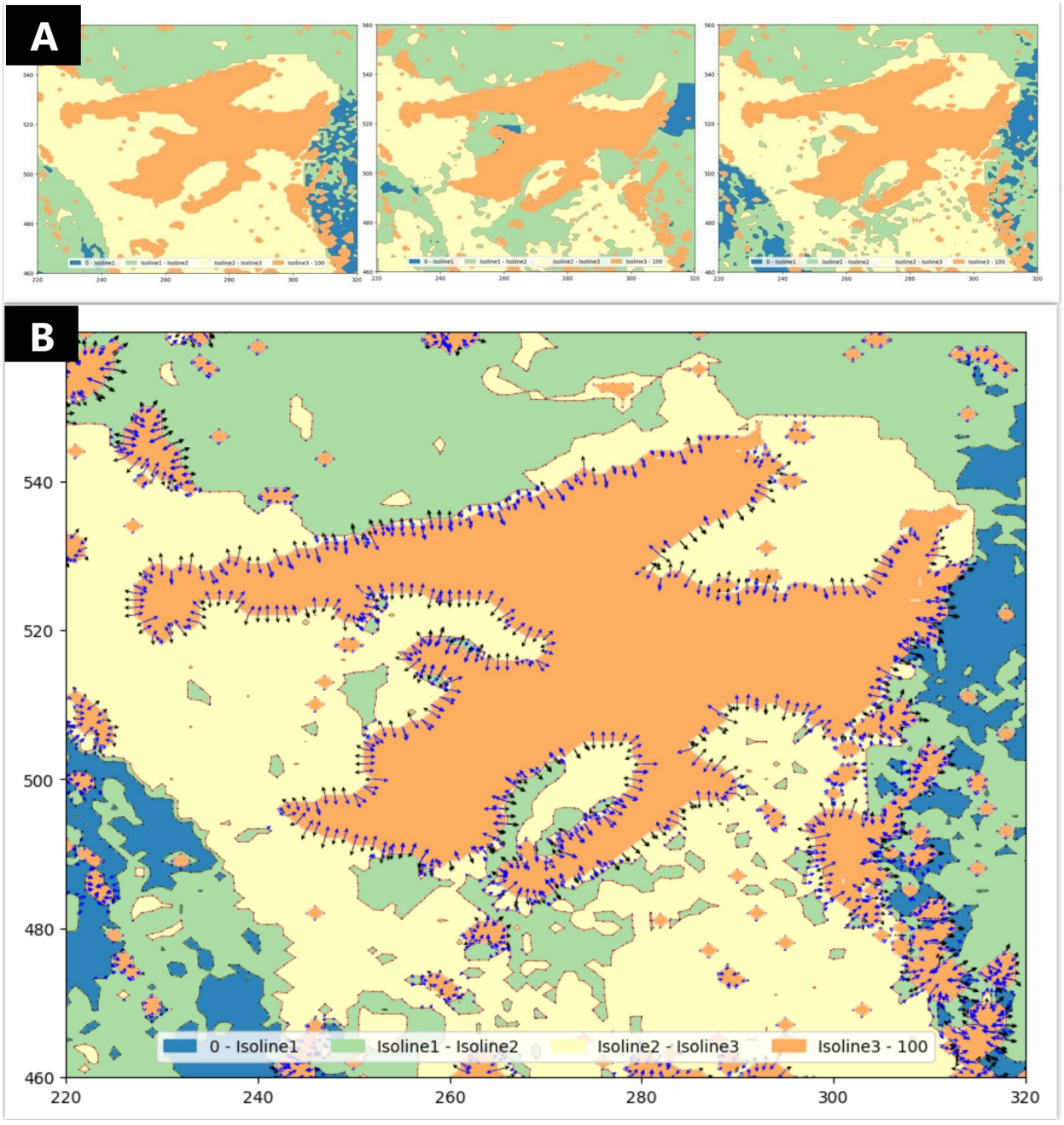

Figure 7.12: (A) Side-by-side contourmaps of Soil Moisture over the Great Bear lake for March 2013 (left), March 2014 (middle), and March 2015 (right); (B) Trend vectors from March 2013 to March 2015 of Soil Moisture along the contours of March 2015.

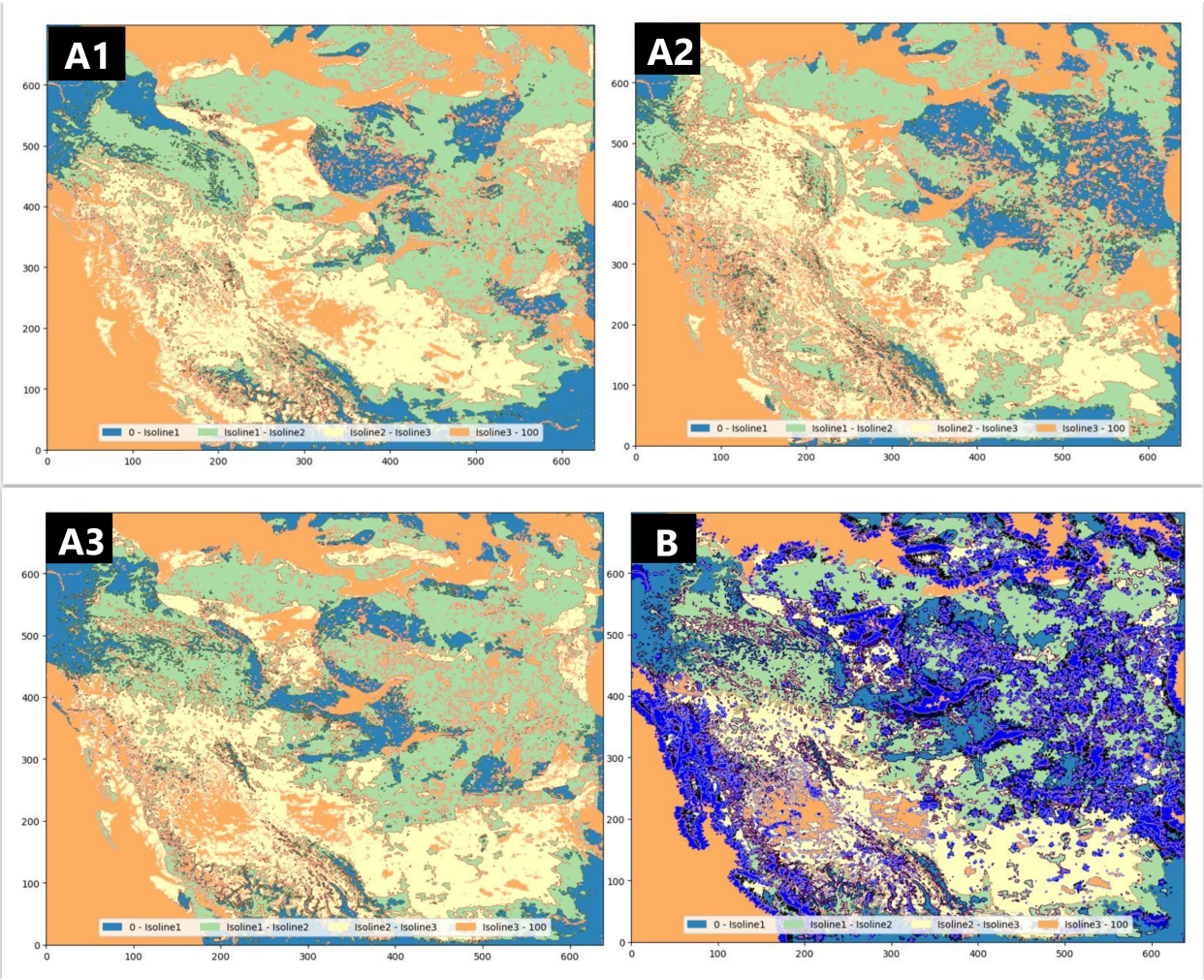

Figure 7.13: Contourmaps of Soil Moisture over Western Canada for (A) March 2013, (B) March 2014, and (C) March 2015; (D) Trend vectors from March 2013 to March 2015 of Soil Moisture along the contours of March 2015.

whole region in Figure 7.13, she commented: *"With this, I find it very hard to read the contours because there are so many changes everywhere"*. So, she focused on the trend vectors to identify the prominent changes in interesting regions. For example, in the left corner, there are two adjacent small islands named 'Graham Island' and 'Moresby Island'. She observed the directions in the island and readily figured out from the trend vectors that the island was getting drier in the Summer, which is consistent with the contourmaps. Overall, the expert felt the larger side-by-side maps constituted (a) a lot of changes and (b) many localized changes, which are strenuous to figure out and our trend vectors can be very helpful as a summary plot for such changes.

- **Task 3** - Since the expert wanted to differentiate between long term changes of two different seasons, we showed her at first, the overall change over Western Canada from Figure 7.14. She identified that in the Fall (7.14B), there is a dry front coming from USA compared to Winter (7.14A). That is why the number of 'blue' arrows increased a lot in the Fall. The expert was curious about the changes in the north coast along the Arctic Ocean. She identified that the 'low to high' changes over Winter is not that aggressive. *"Larger precipitation patterns could be a reason that make the soils wetter in the Fall compared to Winter"*, she commented. However, she felt a trend pattern of a larger time difference of only one season would allow her to find more key insight. Based on previous observations in relation to easeness in looking for changes in zoomed in images, the expert moved to Figures 7.15 and 7.16, representing the Great Bear lake and the Slave lake, respectively. Since these are more localized changes, she noticed more 'blue' arrows than 'black' arrows in 7.15A. She figured that soil moisture in this region is increasing along the edges. But she was amazed by the changes over Fall, shown in 7.15B. Though she noticed more 'blue' arrows, she identified there are certain areas that shows no arrows. After careful consideration, she pointed out that since the lake was getting bigger in the Fall, the changes of a point and its neighboring regions would decrease significantly. Consequently, the increased 'orange' region in this image shows no trend vector at all. The expert did not find any real difference in change patterns between Winter and Fall. She thought that it was interesting and it could introduce further research over longer period of time.

Feedback

After the task scenarios, we interviewed the expert and collected useful feedback and suggestions on our system. Overall, the expert was very pleased with our system and the generated images. She felt that the system was useful and it can provide an overall summary of the changes over time. She commented: *"With quite a bit of training, I found these trend vector maps to be a good summary of trends across multiple time stamps. If I used this tool for my work, I would probably use it as a first diagnostics tool to see an overview of trends over an area. From this trend vectors map, I would then identify smaller regions or interesting patterns that I would then analyze using my own tools"*. She also felt this tool could especially be useful for people working with climate change scenarios. She would also like to analyze various other geospatial variables

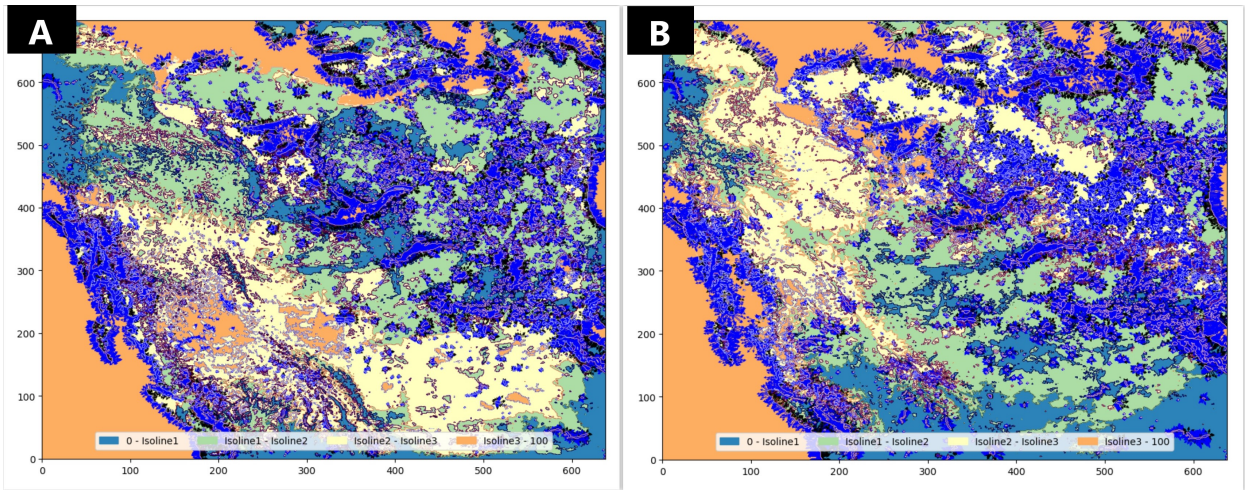

Figure 7.14: Trend vectors of Soil Moisture over Western Canada for (A) March 2013 to March 2015 and (B) September 2013 to September 2015.

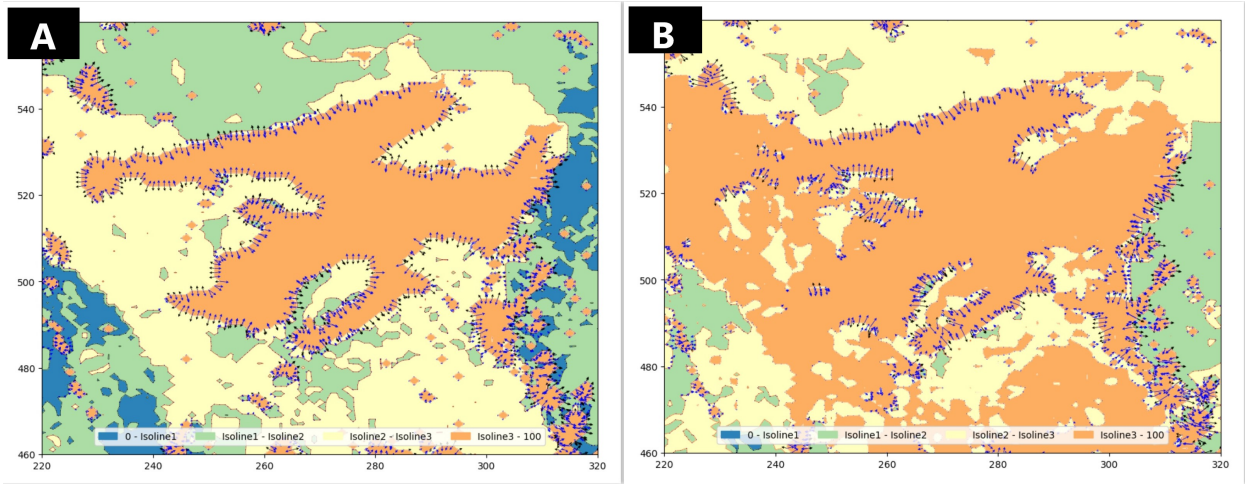

Figure 7.15: Trend vectors of Soil Moisture over the Great Bear lake for (A) March 2013 to March 2015 and (B) September 2013 to September 2015.

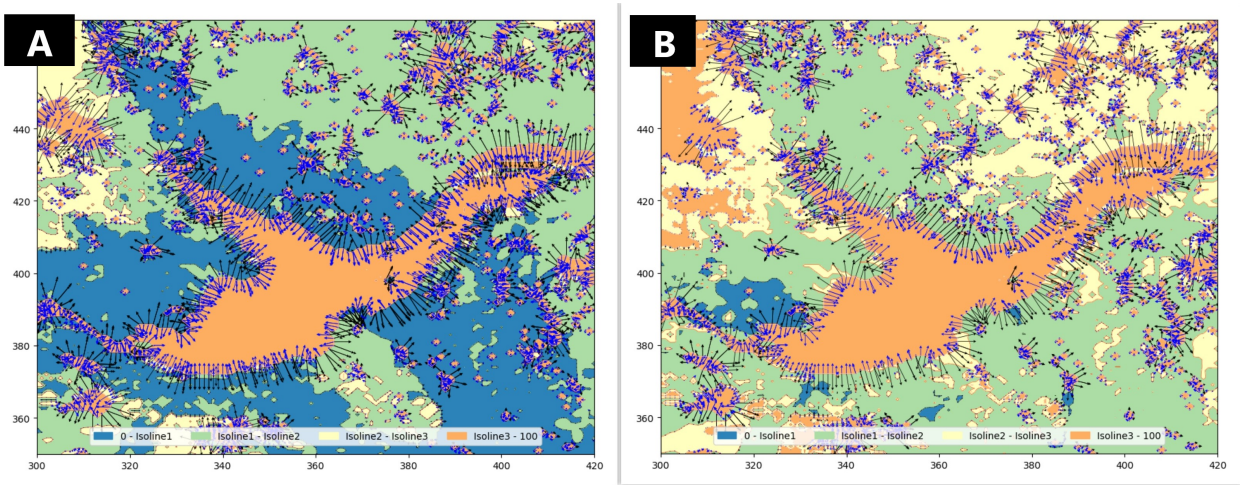

Figure 7.16: Trend vectors of Soil Moisture over the Great Slave lake for (A) March 2013 to March 2015 and (B) September 2013 to September 2015.

such as, snow water equivalent, precipitation, temperature, and ground water. However, she provided two suggestions towards the improvement of overall system. Firstly, since she and most of her colleagues use netcdf files as input, she suggested expanding to other file formats. Additionally, some of the variables are not continuous, and so, they can trickier to handle using our system. Secondly, additional statistical measures such as time series plot of the dataset can be added as a secondary mechanism of evaluation. This can be particularly useful to detect smaller changes over time or a sudden data deviation in multiple timestamps. The expert found the contourmap and vector interactions to be practical and appreciated the overall neatness of our tool design.

7.4 Overall Discussion

Here we discuss the insights obtained from the findings of ContourDiff with respect to the questions we set for our research.

Research Question 1*: How can we visualize spatio-temporal changes in a single and static view? How can this view be analyzed for different change patterns over different time periods?*

We visualize spatio-temporal changes through a framework, ContourDiff, that reveals change patterns through vectors overlaid on top of a contourmap. The framework includes a novel visualization approach that uses gradients to signify changes of a point to its neighbours and shows these changes by arrows along the contour lines. We design the contourmaps as a function of boxplots and color to present a clear picture of the recent most landscape. The magnitude of the vector indicates the change values and the direction points to change movement. Since we find almost similar change directions along the contour lines, we depict this as a *"change trend"* over a time period in that spatial region. Furthermore, we divide these patterns into *'increasing'* and *'decreasing'* categories to provide a more enriched understanding of the changes. Since

a temporal occurrence can be depicted with numerical values or images, our approach considers them as input possibilities as well. We introduce a synthetic dataset for readability into our created visualization. While animation or side-by-side images can be difficult to compare to find small changes or strenuous for human interpretation, ContourDiff produces a single and static change visualization. Hence, users can at first capture an overview, and then get a more localized perspective, if needed. Our expert evaluation reveals that visualizations were useful to reveal interesting pattern from real-life dataset

Research Question 2*: How do we initially extract patterns over the large scale data as fast as possible? How do we provide users maximum query throughput with minimum time latency?*

Our numerical WRF dataset contains a 699×639 number of elements in a single time step with 26 geospatial variables. To extract patterns from such a large scale data, we design a sorted data dictionary and apply sequential processing. We model our dataframe into a graph structure that allows us to access a spatial point as a node and generate weighted graphs easily for one temporal dataset. We also apply a series of modifications to reduce the time overall processing time to construct a ContourDiff image. To avoid user interaction latency, we employ quadtree based data structure to allow users store and access data from contour paths more quickly. This allows significant reduction of time for locating and visualizing vectors while the quadtree depth, magnitude value, or direction of the vectors is changed. We report that our approach executes upto a million data points under a minute. Moreover, we observed from our expert evaluation that side-by-side images take more time to analyze while there are a lot of regions with changes or number of timestamps grow.

Research Question 3*: How to effectively and efficiently visualize curated information from raw data with a well-designed and interactive user interface?*

Our ContourDiff framework includes an interactive platform that enables users navigate through the generated visualization very quickly. The interface elements are designed keeping user requirements in mind, such as, grouping similar features together, providing navigation for synthetic geographical location, employing multiple direct action input mechanism (zooming, panning, editing image parameter, saving images). ContourDiff also accommodates a smaller frame into reading data distribution and manipulating it to create user defined contourmaps. A draft Contourdiff image is curated with basic map and content properties which can be easily changed from the control panel. The main display shows the image with a colormap legend to identify the data percentile and its representation on the image. Icons for direct interaction have been utilized because they are easy to interpret and remember. Zooming and panning are easy to use and efficient in focusing on small, yet, significant changes. A (x, y) cursor helps users with the navigation for overview and zoomed-in images. Additionally, the parameter edit option allows for pinpoint axes configuration with location and label names. Saving a single, static image with user defined modifications may help for future usage and exploration. Once the computation is done, information regarding vector computation for all data points are stored. Hence, the user interactions in ContourDiff are real-time as well.

8 Map Projection

This chapter provides the projection details used in our system. The dataset we used to generate the visualizations are two-dimensional. For example, the Weather Research and Forecasting model output we have used in Chapter 7 to verify our visualization model are .csv files with (x, y) locations and their respective values for a variable v. These .csv files have been extracted from WRF netcdf files. The overall area is Western Canada that has been constructed as a 699×639 grid where each cell of the grid corresponds to a $4km \times 4km$ area. We used the given grid directly to create our contour plot. Hence it is a natural question to ask whether the generated images would look different if a projection technique had been used. Here we show that the original ContourDiff system used a uniform grid that generates visualizations different from the ones that we obtain by equal-area projection as the projection shifts the uniform grid. One limitation of our system is that it currently does not provide support for other common map projections; but assumes a projected set of data has been given as an input.

8.1 Introduction

A map projection is a systematic transformation of the latitudes and longitudes of positions on the surface of the Earth to a flat sheet of paper, a map. More precisely, a map projection requires a transformation from a set of two independent coordinates (latitude ϕ and longitude λ) on the model of the Earth to a set of two independent coordinates (x and y) on a Cartesian representation (the map)[50]. There are a number of techniques for map projection, yet with all of them distortion occurs in angle, shape, or area. Carl Friedrich Gauss showed that a sphere's surface cannot be represented on a map without distortion [112].

The Mercator projection is one of the most well-known projections and has been heavily used in various atlases. Figure 8.1 shows the Mercator projection of the world. It is a conformal projection that preserves shape. Area is distorted as distance from the standard line increases. The main limitation of this projection is that it exaggerates areas far from the equator. For example, Greenland appears the same size as Africa, when in reality Africa's area is 14 times as large [105]. Moreover, Antarctica appears to be extremely large. If the entire globe were mapped, Antarctica would inflate infinitely. In reality, it is the second smallest continent, being just smaller than Russia [105].

So, the selection of an appropriate map projection has a fundamental impact on the visualization and analysis of geographic information [52]. Though these map visualizations have various limitations, useful traits of maps such as accommodation and storing enormous range of values, motivate the development of

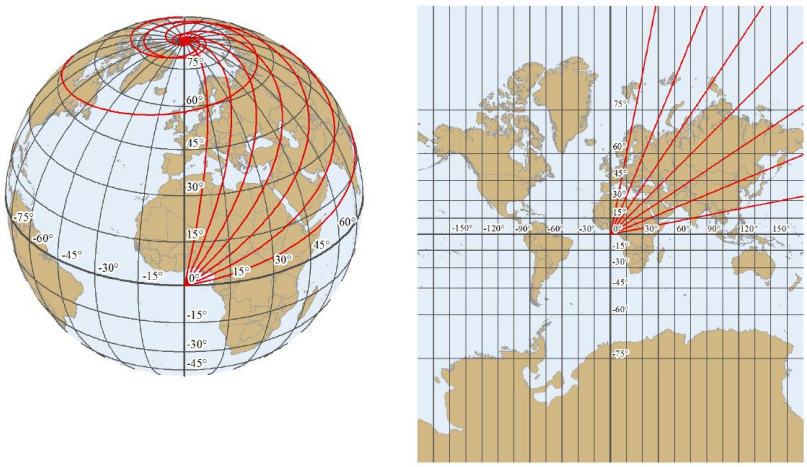

Figure 8.1: (Left) The Globe and, (Right) Mercator Projection [149]

map projections.

8.2 WRF Map Projection

The WRF files we used were developed based on the Lambert Conformal Conic projection. It is a conformal projection like Mercator, but reduces area distortion by using two standard parallels [5]. It can also be defined with a single standard parallel and a scale factor [5]. It is best suited for conformal mapping of land masses extending in an east-to-west orientation at mid-latitudes or with an orientation parallel to the standard lines. [5]. Directions, angles, and shapes are maintained at the standard lines but they are increasingly distorted away from these lines [5]. The WRF default terrain and land-cover data are all provided in the standard WGS84 Datum.

Figure 8.2 illustrates how WRF coordinate transformation is performed. As explained by [6], the method is as follows:

- A classic transformation between two map projections requires three steps: (1) the conversion of eastings/northings to lons/lats, (2) a datum shift, and (3) convert back the lons/lats to eastings/northings
- One has to use the radius of the WRF sphere to define the WRF map projection (in this case Lambert Conformal Conic, but it's also true for Mercator or Polar projections)
- The lon/lat coordinates obtained from the conversion of the spherical WRF eastings/northings need to be interpreted as being in the standard WGS84 datum, not as spherical lon/lat. Therefore no datum shift is required

Although a number of map projections can be used, we here use equal area projection and compare the visualizations with the ones from Chapter 7.

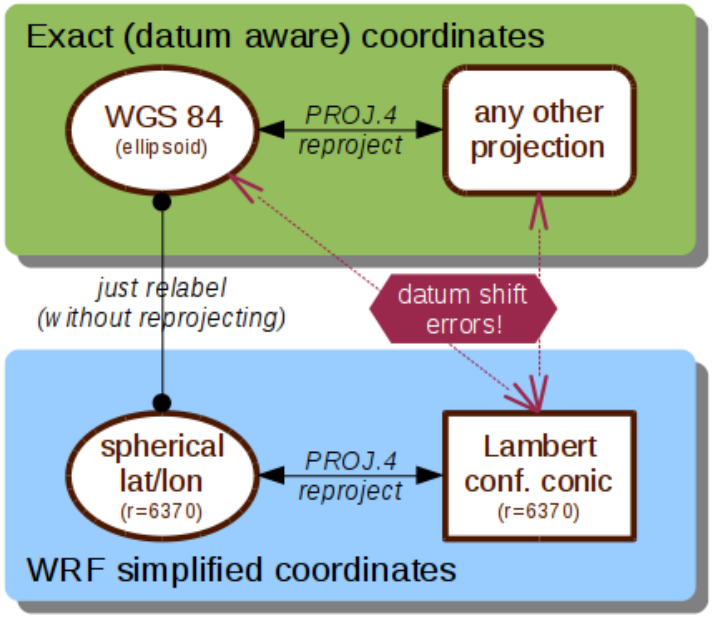

Figure 8.2: Coordinate transformation diagram in WRF Projection [6]

8.3 Equal Area Projection

The equal-area projection retains the relative size of the area throughout a map. So that means at any given region in a map, an equal-area projection keeps the true size of features. For example, on an equal-area world map, Norway takes up the same percentage of map space that actual Norway takes up on the earth. While equal-area projections preserve area, they distort shape and angles and so are not conformal [3]. The equator is chosen as the parallel of no distortion, but distortion increases rapidly towards the poles [153]. Behrmann projection is an example of cylindrical equal-area projection, with 30°N and 30°S as parallel of distortion. Gall-Peters has the parallel of distortion as 45°N and 45°S.

8.4 Procedure

Here the step-by-step computation for equal-area projection and subsequent visualization has been described. This computation workflow has been explained with Figure 8.3.

- We first transform each (x, y) point of the $(699, 639)$ matrix of our WRF dataset to it's respective $(latitude, longitude)$. The equations to do this transformation are below:

$$\phi_2 = asin(sin\phi_1 \times cos\delta + cos\phi_1 \times sin\delta \times cos\theta) \tag{8.1}$$

$$\lambda_2 = \lambda_1 + atan2(sin\theta \times sin\delta \times cos\phi_1, cos\delta - sin\phi_1 \times sin\phi_2) \tag{8.2}$$

where (ϕ_1, λ_1) are center points, extracted from WRF netcdf files, which is $(60, -117)$. θ is the bearing (clockwise from north), δ is the angular distance d/R; d being the distance travelled, R the earth's radius (6371 km). (ϕ_2, λ_2) are the calculated latitude and longitude points.

- Then we are projecting these points with equal-area projection. We utilize Matlab's grn2eqa function [1] to calculate the projected (x, y) coordinates from previously computed latitude and longitude degrees.
- Then, we transform these (x, y) coordinates into screen coordinates $(699, 639)$. For the x value transformation, $(x - x.min) \times 699/|x.max - x.min|$ and y value transformation, $(y - y.min) \times 639/|y.max - y.min|$ is utilized.
- For each unique screen coordinates computed in the last step, we obtain the value of the variable from original WRF netcdf file. For other positions, we calculate the average of the values that exist around the 5×5 square of the position.
- Finally we get a structure of [screen x coordinate, screen y coordinate, value] that is transformed as .csv file and taken as the input of our ContourDiff system.

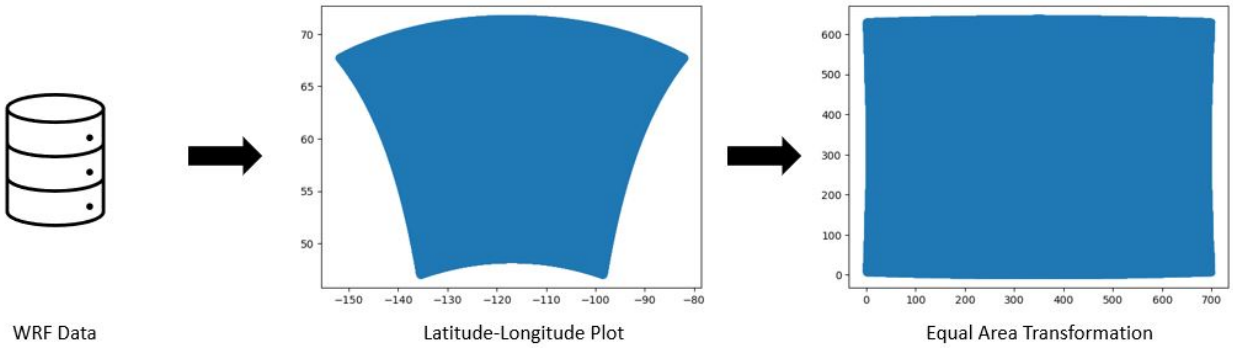

Figure 8.3: Computation of equal-area projection from WRF data

8.5 Result and Comparison

Here we show the resultant visualizations computed from our projected input and compare them against the generated ones from Chapter 7. The left image in Figure 8.4 shows the original visualization of Western Canada for Soil Moisture in January 2013 compared to projected visualization on the right. The left image in Figure 8.5 shows the original visualization of The Great Slave Lake for Soil Moisture in January 2013 compared to projected visualization on the right. Finally, the left image in Figure 8.6 shows the original visualization of The Great Bear Lake for Soil Moisture in January 2013 compared to projected visualization on the right.

The projection used in Chapter 7 is based on a uniform grid and so those visualizations differ from the ones we produce using the equal-area projection. However as shown in Figures 8.4, 8.5, and 8.6, the differences are not readily visible.

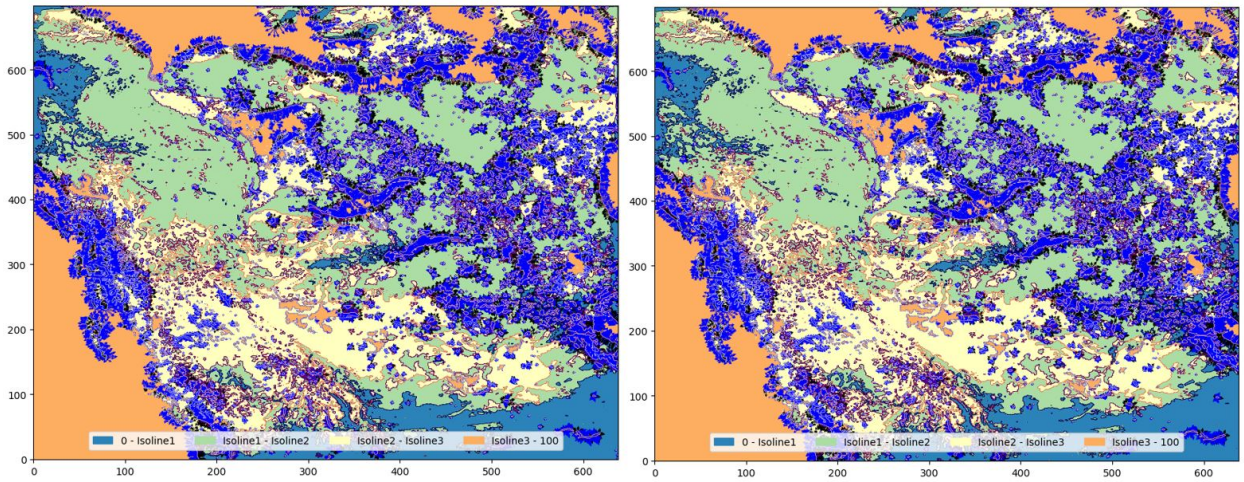

Figure 8.4: (left) ContourDiff generated image for original data, (right) ContourDiff generated image for projected data over Western Canada.

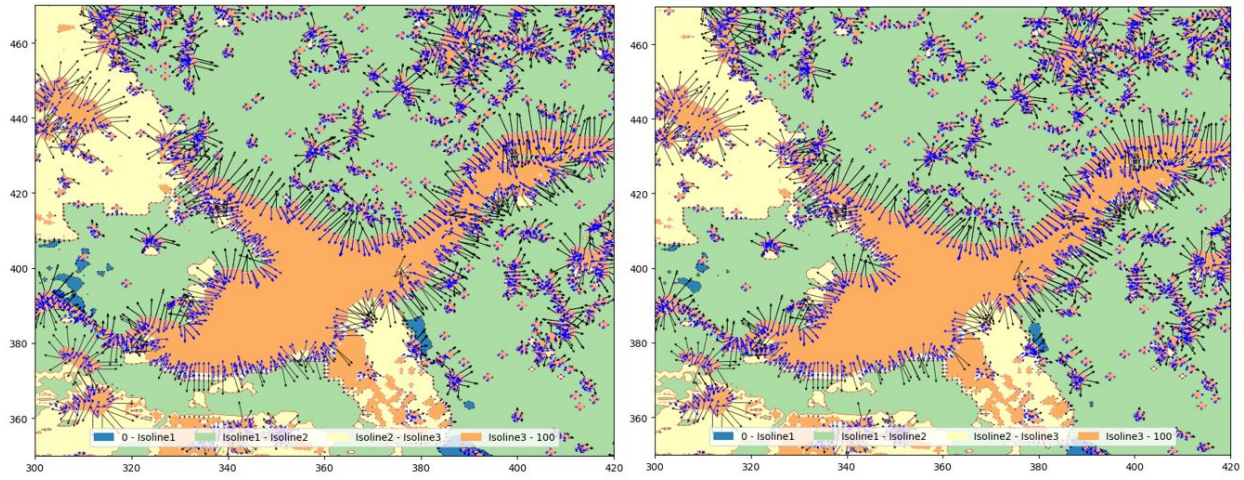

Figure 8.5: (left) ContourDiff generated image for original data, (right) ContourDiff generated image for projected data over The Great Slave Lake.

To more clearly show the differences in the visualizations, we first generate contour maps for both the original data and the new projected data for a region (Figure 8.7) and then zoom in on two subregions, region A (Figure 8.8) and region B (Figure 8.9).

For zoomed-in top Region A, we can see from Figure 8.8 how the projection shifts the uniform grid in the Y-axis. This shift is more visible towards the north pole, but increasingly more uniform in the central part of the visualization. For example, the shift of zoomed-in lower Region B in Figure 8.9 is smaller than the shift we see in Region A.

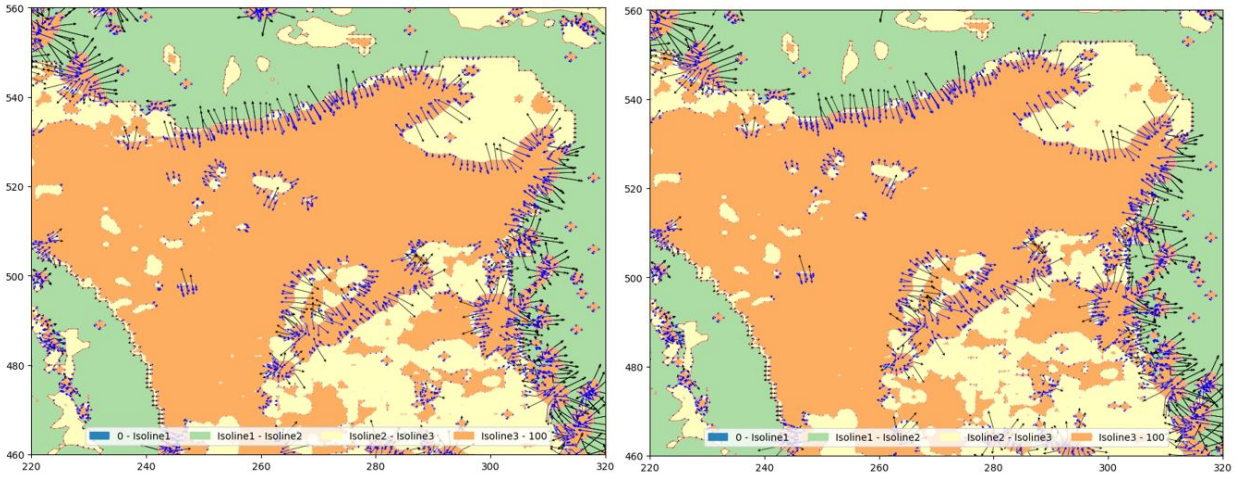

Figure 8.6: (left) ContourDiff generated image for original data, (right) ContourDiff generated image for projected data over The Great Bear Lake.

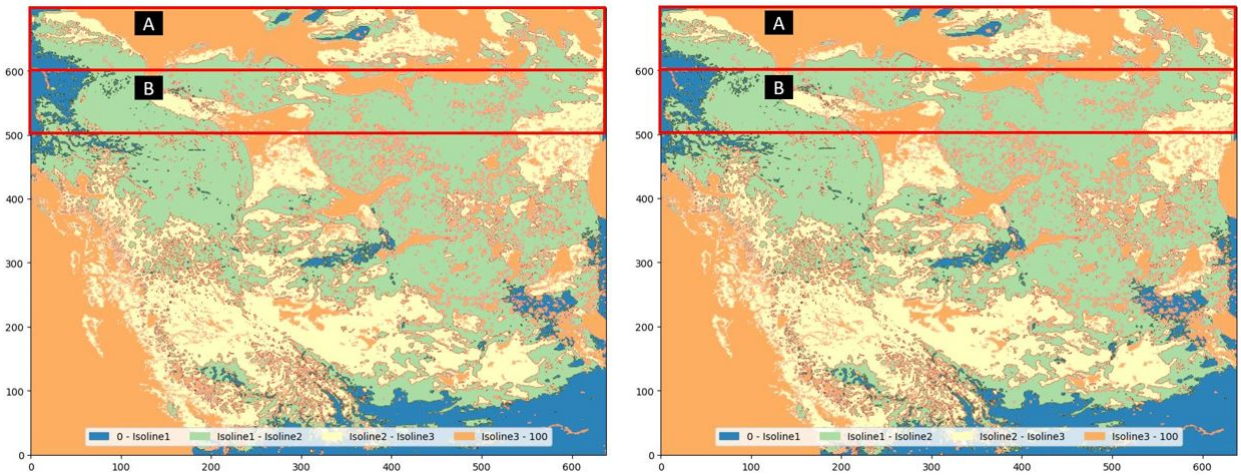

Figure 8.7: Region A and Region B of Contourmap for (left) original data and (right) projected data.

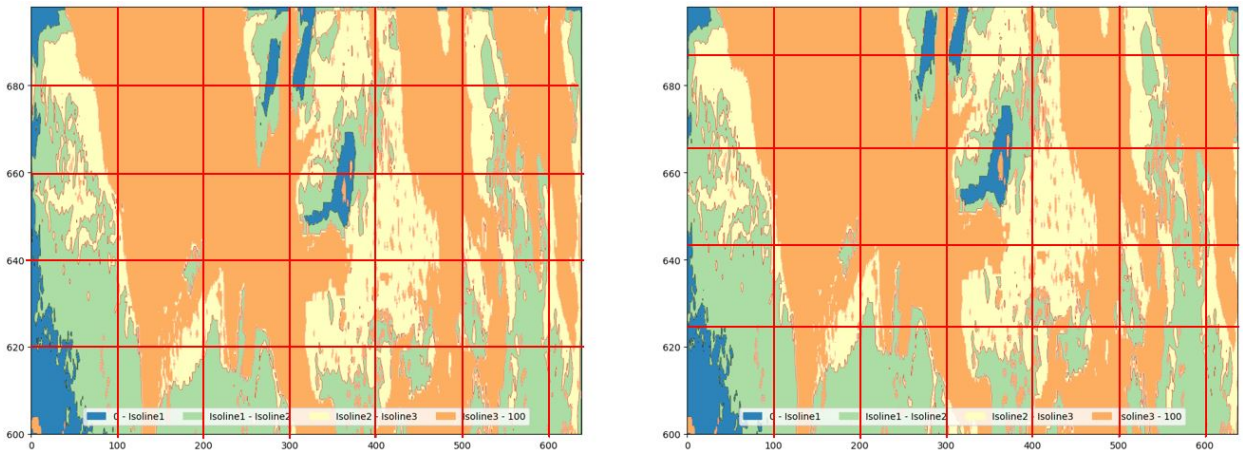

Figure 8.8: (left) Zoomed-in visualization of region A with uniform grid, (right) The grid shifts with equal-area projection towards the north pole.

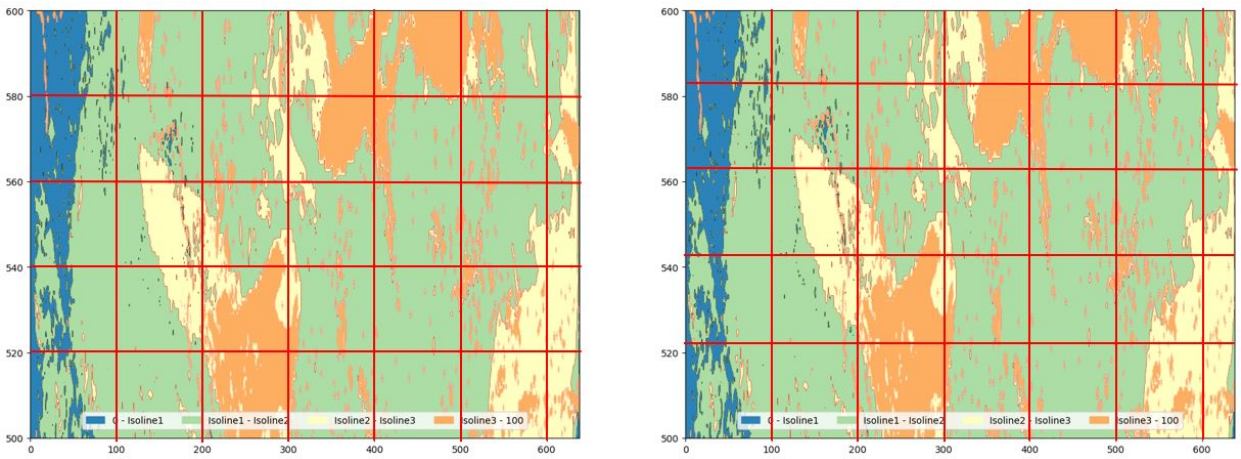

Figure 8.9: (left) Zoomed-in visualization of region B with uniform grid, (right) The grid shift is smaller than the shift in region A.

8.6 Summary

We can utilize other projections based on the steps that we have used to compute projected screen coordinates. The resultant images might differ based on other projections. But these projections are very important to adapt ContourDiff for the system to provide users with choices for proper geological analysis.

9 Conclusion

Representation of spatio-temporal data is of importance since geo-referenced data over time is growing and analysis of these data yields valuable insight about an event. But standard techniques are limited in providing intuitive understanding from change visualization. To address this issue, we explore a framework that provides a summarization of these changes over time. This framework combines a novel vector-based visualization approach and a system for user interaction.

9.1 Contributions

Here we discuss about the three primary contribution presented in this thesis.

- **Vector as a representation of change:** We show that vectors can used to identify and summarize changes. We utilize the relation between a data point and its neighbours to discover the change direction even though the dataset is not directional. A combination of vectors along with colored contourmaps allows finding trend patterns over a period of time. This is beneficial compared to statistical summary plot that is visually challenging for users to understand the patterns correctly for a large dataset and animation, or side-by-side plot that can lead to difficulties in perception of change in distant timestamps. The vectors also provide accurate representation of change from real-life phenomena, as expert evaluations and motion vectors suggest.

- **Fast and scalable visualization:** We use large-scale, real-life dataset to understand change over time. Our visualization approach consists of multiple strategies such as graph generation, quadtree implementation, etc. to make it fast and scalable. Storing and sorting information with quadtree help us visualize information and answer user queries regarding magnitude of change, type of trends and image resolution effectively and efficiently.

- **User interface:** We design and employ a user interface with the visualization approach, for real-time interactions. This helps generating changes and navigating through them really easy. A high interactivity with the system helps users utilize mechanism such as zooming and panning, interchange between states of the visualization keeps them interested and informed in minimal time. As one of experts suggested, people working with geospatial variables can use this as a first diagnostic tool with a bit of training.

9.2 Limitations and Future Work

Our experience with the overall system indicates some limitations in the research that provide opportunities for additional study.

- **GPU implementation:** We have reported that our system takes under 1 minute to process 1 million data points. That can be a big problem in the future if users want to try a larger dataset. Since the dataset is processed in Pandas and vectors are computed from large matrices in Numpy of Python language, even faster CPUs do not have enough processing power to handle it. Bergstra et al. [22] reported that a program of large mathematical expression including hundreds of elementary operations can be a bottleneck for program and performance optimization. So, it would be interesting to use GPU (Graphics Processing Unit) to leverage its parallel processing capability in our system. Even though there is some overhead time to transfer data from CPU to GPU, the total computation time can be significantly reduced for larger datasets.

- **Additional system configurations:** Currently the system can only take csv files and image files as input. But our evaluator suggested netcdf files as a input type in the system. Additionally, in the future, we would like to work with discrete dataset rather than continuous, matrix data. Secondly, statistical overviews can be placed in the system for users to understand data characteristics or compare them before plotting. Moreover, a linked view can be provided with textual and statistical plot (e.g., time-series plots) of the change information.

- **Usability study:** A future avenue for research is to explore various domains by adding a number of subject systems. Additionally, we can conduct a usability study for our interface to find bottlenecks and improve the current system. A combination of broad-spectrum test cases with user feedback can help with universality of our system in interdisciplinary areas.

9.3 Summary

We propose ContourDiff, a vector-based visualization approach that overlays vectors over contour map to analyze the trends of change across spatial regions and temporal domain. We also provide an interactive system for the users to explore ContourDiff visualization. To reduce interaction time, we store vector values of the observed topological area implementing quadtree technique. We used Weather Research and Forecasting (WRF) model output to evaluate our system for real-life geospatial parameters and were able to revel trends that are consistent with the existing knowledge about these geospatial parameters. We also used image data to show how ContourDiff visualization can be used to reveal subtle changes between almost identical images. We believe the approach will inspire future research on visualizing differential trends in geospatial data and changes in timelapse images.

References

[1] Convert from greenwich to equal area coordinates. https://www.mathworks.com/help/map/ref/grn2eqa.html.

[2] Definition of the nhc track forecast cone. https://www.nhc.noaa.gov/aboutcone.shtml.

[3] Equal area projection maps advantages and examples. https://gisgeography.com/equal-area-projection-maps/.

[4] Generate isotropic gaussian and label samples by quantile. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_gaussian_quantiles.html.

[5] Lambert conformal conic. https://desktop.arcgis.com/en/arcmap/latest/map/projections/lambert-conformal-conic.html.

[6] The pitfalls of lambert conformal projection in wrf and mm5. http://www.pkrc.net/wrf-lambert.html.

[7] The pr quadtree. https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/PRquadtree.html.

[8] Water movement in soils. http://www.soilphysics.okstate.edu/software/water/infil.html.

[9] Jae-wook Ahn, Meirav Taieb-Maimon, Awalin Sopan, Catherine Plaisant, and Ben Shneiderman. Temporal visualization of social network dynamics: Prototypes for nation of neighbors. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 309–316. Springer, 2011.

[10] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visualizing time-oriented data—a systematic view. *Computers & Graphics*, 31(3):401–409, 2007.

[11] Wolfgang Aigner, Silvia Miksch, Bettina Thurnher, and Stefan Biffl. Planninglines: novel glyphs for representing temporal uncertainties and their evaluation. In *Ninth International Conference on Information Visualisation (IV'05)*, pages 457–463. IEEE, 2005.

[12] Gennady Andrienko, Natalia Andrienko, Martin Mladenov, Michael Mock, and Christian Politz. Identifying place histories from activity traces with an eye to parameter impact. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):675–688, 2011.

[13] Gennady Andrienko, Natalia Andrienko, and Stefan Wrobel. Visual analytics tools for analysis of movement data. *ACM SIGKDD Explorations Newsletter*, 9(2):38–46, 2007.

[14] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Visualization of spatio-temporal information in the internet. In *Proceedings 11th International Workshop on Database and Expert Systems Applications*, pages 577–585. IEEE, 2000.

[15] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Exploring changes in census time series with interactive dynamic maps and graphics. *Computational statistics*, 16(3):417–433, 2001.

[16] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503–541, 2003.

[17] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Tools for visual comparison of spatial development scenarios. In *Proceedings on Seventh International Conference on Information Visualization, 2003. IV 2003.*, pages 237–244. IEEE, 2003.

[18] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Impact of data and task characteristics on design of spatio-temporal data visualization tools. In *Exploring geovisualization*, pages 201–222. Elsevier, 2005.

[19] Natalia Andrienko, Gennady Andrienko, Robert Peckham, and Hans Voss. Commongis: Common access to geographically referenced data. In *Abstracts of 6th EC-GIS Workshop*, pages 59–60. European Commission Joint Research Centre, 2000.

[20] Daniel Archambault, Helen Purchase, and Bruno Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE transactions on visualization and computer graphics*, 17(4):539–552, 2010.

[21] Cornelia Auer, Jens Kasten, Andrea Kratz, Eugene Zhang, and Ingrid Hotz. Automatic, tensor-guided illustrative vector field visualization. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 265–272. IEEE, 2013.

[22] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, volume 1, pages 3–10, 2010.

[23] Serge Beucher et al. The watershed transformation applied to image segmentation. *Scanning microscopy-supplement-*, pages 299–299, 1992.

[24] Connie Blok. Monitoring change: characteristics of dynamic geo-spatial phenomena for visual exploration. In *Spatial cognition II*, pages 16–30. Springer, 2000.

[25] David Borland and Russell M Taylor Ii. Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 27(2):14–17, 2007.

[26] Matthias Bürgi, Anna M Hersperger, and Nina Schneeberger. Driving forces of landscape change-current and new directions. *Landscape ecology*, 19(8):857–868, 2005.

[27] Iain Bruce Campbell and GGC Claridge. *Antarctica: soils, weathering processes and environment*, volume 16. Elsevier, 1987.

[28] Nan Cao, Chaoguang Lin, Qiuhan Zhu, Yu-Ru Lin, Xian Teng, and Xidao Wen. Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data. *IEEE transactions on visualization and computer graphics*, 24(1):23–33, 2017.

[29] William Cartwright, Jeremy Crampton, Georg Gartner, Suzette Miller, Kirk Mitchell, Eva Siekierska, and Jo Wood. Geospatial information visualization user interface issues. *Cartography and Geographic Information Science*, 28(1):45–60, 2001.

[30] Bevan L Cheeseman, Ulrik Günther, Krzysztof Gonciarz, Mateusz Susik, and Ivo F Sbalzarini. Adaptive particle representation of fluorescence microscopy images. *Nature communications*, 9(1):1–13, 2018.

[31] Hsinchun Chen, Homa Atabakhsh, Chunju Tseng, Byron Marshall, Siddharth Kaza, Shauna Eggers, Hemanth Gowda, Ankit Shah, Tim Petersen, and Chuck Violette. Visualization in law enforcement. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1268–1271, 2005.

[32] Xueyi Chen, Liming Shen, Ziqi Sha, Richen Liu, Siming Chen, Genlin Ji, and Chao Tan. A survey of multi-space techniques in spatio-temporal simulation data visualization. *Visual Informatics*, 3(3):129–139, 2019.

[33] S Chick, PJ Sánchez, D Ferrin, and DJ Morrice. Visualization methods for time-dependent data-an overview. In *Proceedings of the 2003 Winter Simulation Conference*. Citeseer, 2003.

[34] Kristin Cook, Georges Grinstein, Mark Whiting, Michael Cooper, Paul Havig, Kristen Liggett, Bohdan Nebesh, and Celeste Lyn Paul. Vast challenge 2012: Visual analytics for big data. In *2012 IEEE conference on visual analytics science and technology (VAST)*, pages 251–255. IEEE, 2012.

[35] Jonathan Cox, Donald House, and Michael Lindell. Visualizing uncertainty in predicted hurricane tracks. *International Journal for Uncertainty Quantification*, 3(2), 2013.

[36] Andrea Cuttone, Sune Lehmann, and Jakob Eg Larsen. geoplotlib: a python toolbox for visualizing geographical data. *arXiv preprint arXiv:1608.01933*, 2016.

[37] Caitlin Dempsey. Where is the phrase "80% of data is geographic" from?'. *GIS Lounge*, 28, 2012.

[38] Urška Demšar and Kirsi Virrantaus. Space–time density of trajectories: exploring spatio-temporal patterns in movement data. *International Journal of Geographical Information Science*, 24(10):1527–1542, 2010.

[39] Alexandra Diehl, Leandro Pelorosso, Claudio Delrieux, Celeste Saulo, Juan Ruiz, M Eduard Gröller, and Stefan Bruckner. Visual analysis of spatio-temporal data: Applications in weather forecasting. In *Computer Graphics Forum*, volume 34, pages 381–390. Wiley Online Library, 2015.

[40] Peter Diggle, Peter J Diggle, Patrick Heagerty, Patrick J Heagerty, Kung-Yee Liang, Scott Zeger, et al. *Analysis of longitudinal data*. Oxford University Press, 2002.

[41] Alan Dix and Geoff Ellis. By chance enhancing interaction with large data sets through statistical sampling. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 167–176, 2002.

[42] Helmut Doleisch, Michael Mayer, Martin Gasser, Roland Wanker, and Helwig Hauser. Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system. In *VisSym*, pages 91–96. Citeseer, 2004.

[43] Jerry Alan Fails, Amy Karlson, Layla Shahamat, and Ben Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 167–174. IEEE, 2006.

[44] Florian Ferstl, Mathias Kanzler, Marc Rautenhaus, and Rüdiger Westermann. Time-hierarchical clustering and visualization of weather forecast ensembles. *IEEE transactions on visualization and computer graphics*, 23(1):831–840, 2016.

[45] Alexey Fofonov and Lars Linsen. Projected field similarity for comparative visualization of multi-run multi-field time-varying spatial data. In *Computer Graphics Forum*, volume 38, pages 286–299. Wiley Online Library, 2019.

[46] Alexey Fofonov, Vladimir Molchanov, and Lars Linsen. Visual analysis of multi-run spatio-temporal simulations using isocontour similarity for projected views. *IEEE transactions on visualization and computer graphics*, 22(8):2037–2050, 2015.

[47] Felice Frankel and Rosalind Reid. Big data: Distilling meaning from data. *Nature*, 455(7209):30–30, 2008.

[48] Peter Gatalsky, Natalia Andrienko, and Gennady Andrienko. Interactive analysis of event data using space-time cube. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, pages 145–152. IEEE, 2004.

[49] Jim Gemmell, Gordon Bell, and Roger Lueder. Mylifebits: a personal database for everything. *Communications of the ACM*, 49(1):88–95, 2006.

[50] Ebrahim Ghaderpour. Some equal-area, conformal and conventional map projections: A tutorial review. *Journal of Applied Geodesy*, 10(3):197–209, 2016.

[51] Rafael Gonzalez and Richard Woods. *Digital Image Processing*. Pearson Education, Inc., 2008.

[52] Paul Christopher Gosling and Elias Symeonakis. Automated map projection selection for gis. *Cartography and Geographic Information Science*, 47(3):261–276, 2020.

[53] Alexander Gruber, C-H Su, Simon Zwieback, W Crow, Wouter Dorigo, and Wolfgang Wagner. Recent advances in (soil moisture) triple collocation analysis. *International Journal of Applied Earth Observation and Geoinformation*, 45:200–211, 2016.

[54] Yi Gu and Chaoli Wang. Transgraph: Hierarchical exploration of transition relationships in time-varying volumetric data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2015–2024, 2011.

[55] Diansheng Guo, Jin Chen, Alan M MacEachren, and Ke Liao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE transactions on visualization and computer graphics*, 12(6):1461–1474, 2006.

[56] Torsten Hagerstrand. What about people in regional. *Papers of the Regional Science Association*, 24(1):7–21, 1970.

[57] Lihua Hao, Christopher G Healey, and Steffen A Bass. Effective visualization of temporal ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):787–796, 2015.

[58] Mark Harrower and Cynthia A Brewer. Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[59] Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: Visualizing theme changes over time. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, pages 115–123. IEEE, 2000.

[60] Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE transactions on visualization and computer graphics*, 8(1):9–20, 2002.

[61] Christopher G Healey and James T Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE transactions on visualization and computer graphics*, 5(2):145–167, 1999.

[62] Tomislav Hengl, Pierre Roudier, Dylan Beaudette, Edzer Pebesma, et al. plotkml: Scientific visualization of spatio-temporal data. *Journal of Statistical Software*, 63(5):1–25, 2015.

[63] K Priyantha Hewagamage, Masahito Hirakawa, and Tadao Ichikawa. Interactive visualization of spatiotemporal patterns using spirals on a geographical map. In *Proceedings 1999 IEEE Symposium on Visual Languages*, pages 296–303. IEEE, 1999.

[64] William L Hibbard, Brian E Paul, David A Santek, Charles R Dyer, Andre L Battaiola, and M-F Voidrot-Martinez. Interactive visualization of earth and space science computations. *Computer*, 27(7):65–72, 1994.

[65] Marcel Hlawatsch, Philipp Leube, Wolfgang Nowak, and Daniel Weiskopf. Flow radar glyphs—static visualization of unsteady flow with uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1949–1958, 2011.

[66] David Hoffer. What does big data look like? visualization is key for humans. *January. Http://Www. Wired. Com/Insights/2014/01/Big-Data-Look-like-Visualization-K*, 2014.

[67] Thomas Höllt, Ahmed Magdy, Guoning Chen, Ganesh Gopalakrishnan, Ibrahim Hoteit, Charles D Hansen, and Markus Hadwiger. Visual analysis of uncertainties in ocean forecasts for planning and operation of off-shore structures. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 185–192. IEEE, 2013.

[68] Thomas Höllt, Ahmed Magdy, Peng Zhan, Guoning Chen, Ganesh Gopalakrishnan, Ibrahim Hoteit, Charles D Hansen, and Markus Hadwiger. Ovis: A framework for visual analysisof ocean forecast ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1114–1126, 2014.

[69] Lorenz Hurni, Hans-Rudolf Bär, and René Sieber. The atlas of switzerland as an interactive multimedia atlas information system. In *Multimedia cartography*, pages 99–112. Springer, 1999.

[70] Satoshi Ide and Julie Maury. Seismic moment, seismic energy, and source duration of slow earthquakes: Application of brownian slow earthquake model to three major subduction zones. *Geophysical Research Letters*, 45(7):3059–3067, 2018.

[71] Masahiko Itoh, Masashi Toyoda, and Masaru Kitsuregawa. An interactive visualization framework for time-series of web graphs in a 3d environment. In *2010 14th International Conference Information Visualisation*, pages 54–60. IEEE, 2010.

[72] Navid Jadidoleslam, Radoslaw Goska, Ricardo Mantilla, and Witold F Krajewski. Hydrovise: A non-proprietary open-source software for hydrologic model and data visualization and evaluation. *Environmental Modelling & Software*, 134:104853, 2020.

[73] Michaela Jarema, Johannes Kehrer, and Rüdiger Westermann. Comparative visual analysis of transport variability in flow ensembles. *Journal of WSCG*, 24(1):25–34, 2016.

[74] Mihaela Jarema, Ismail Demir, Johannes Kehrer, and Rüdiger Westermann. Comparative visual analysis of vector field ensembles. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 81–88. IEEE, 2015.

[75] David Kao, Jennifer L Dungan, and Alex Pang. Visualizing 2d probability distributions from eos satellite image-derived data sets: A case study. In *Proceedings of the conference on Visualization'01*, pages 457–460. IEEE Computer Society, 2001.

[76] David Kao, Alison Luo, Jennifer L Dungan, and Alex Pang. Visualizing spatially varying distribution data. In *Proceedings Sixth International Conference on Information Visualisation*, pages 219–225. IEEE, 2002.

[77] Thomas Kapler and William Wright. Geotime information visualization. *Information visualization*, 4(2):136–146, 2005.

[78] Johannes Kehrer, Philipp Muigg, Helmut Doleisch, and Helwig Hauser. Interactive visual analysis of heterogeneous scientific data across an interface. *IEEE Transactions on Visualization and Computer Graphics*, 17(7):934–946, 2010.

[79] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. Visual analytics: Scope and challenges. In *Visual data mining*, pages 76–90. Springer, 2008.

[80] Oliver Kersting and Jürgen Döllner. Interactive 3d visualization of vector data in gis. In *Proceedings of the 10th ACM international symposium on Advances in geographic information systems*, pages 107–112. ACM, 2002.

[81] Seokyeon Kim, Seongmin Jeong, Insoo Woo, Yun Jang, Ross Maciejewski, and David S Ebert. Data flow analysis and visualization for spatiotemporal statistical data without trajectory information. *IEEE transactions on visualization and computer graphics*, 24(3):1287–1300, 2017.

[82] Alexandra Koussoulakou and Menno-Jan Kraak. Spatia-temporal maps and cartographic communication. *The cartographic journal*, 29(2):101–108, 1992.

[83] Menno-Jan Kraak. The space-time cube revisited from a geovisualization perspective. In *Proc. 21st International Cartographic Conference*, pages 1988–1996. Citeseer, 2003.

[84] Sopan Kurkute, Zhenhua Li, Yanping Li, and Fei Huo. Assessment and projection of the water budget over western canada using convection-permitting weather research and forecasting simulations. *Hydrology and Earth System Sciences*, 24(7):3677–3697, 2020.

[85] Songnian Li, Suzana Dragicevic, Francesc Antón Castro, Monika Sester, Stephan Winter, Arzu Coltekin, Christopher Pettit, Bin Jiang, James Haworth, Alfred Stein, et al. Geospatial big data handling theory and methods: A review and research challenges. *ISPRS journal of Photogrammetry and Remote Sensing*, 115:119–133, 2016.

[86] Lauro Lins, James T Klosowski, and Carlos Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2456–2465, 2013.

[87] Jiyuan Liu, Mingliang Liu, Dafang Zhuang, Zengxiang Zhang, and Xiangzheng Deng. Study on spatial pattern of land-use change in china during 1995–2000. *Science in China Series D: Earth Sciences*, 46(4):373–384, 2003.

[88] Le Liu, Mahsa Mirzargar, Robert M Kirby, Ross Whitaker, and Donald H House. Visualizing time-specific hurricane predictions, with uncertainty, from storm path ensembles. In *Computer Graphics Forum*, volume 34, pages 371–380. Wiley Online Library, 2015.

[89] Richen Liu, Hanqi Guo, and Xiaoru Yuan. User-defined feature comparison for vector field ensembles. *Journal of visualization*, 20(2):217–229, 2017.

[90] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430. Wiley Online Library, 2013.

[91] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG)*, 33(6):1–13, 2014.

[92] IF Vega Lopez, Richard T Snodgrass, and Bongki Moon. Spatiotemporal aggregate computation: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):271–286, 2005.

[93] VLK Lou, TE Mitchell, and Arthur H Heuer. Graphical displays of the thermodynamics of high-temperature gas-solid reactions and their application to oxidation of metals and evaporation of oxides. *Journal of the American Ceramic Society*, 68(2):49–58, 1985.

[94] Linlin Lu and Huadong Guo. Visualization of a digital elevation model. *Data Science Journal*, 6:S481–S484, 2007.

[95] Bo Ma and Alireza Entezari. An interactive framework for visualization of weather forecast ensembles. *IEEE transactions on visualization and computer graphics*, 25(1):1091–1101, 2018.

[96] Alan M MacEachren. *How maps work: representation, visualization, and design*. Guilford Press, 2004.

[97] Alan M MacEachren and Menno-Jan Kraak. Research challenges in geovisualization. *Cartography and geographic information science*, 28(1):3–12, 2001.

[98] Arien Mack, Irvin Rock, et al. *Inattentional blindness*. MIT press, 1998.

[99] Carsten Maple. Geometric design and space planning using the marching squares and marching cube algorithms. In *2003 international conference on geometric modeling and graphics, 2003. Proceedings*, pages 90–95. IEEE, 2003.

[100] Robert McGill, John W Tukey, and Wayne A Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.

[101] Emmett McQuinn, Amit Chourasia, Jürgen P Schulze, and Jean-Bernard Minster. Glyphsea: visualizing vector fields. In *Visualization and Data Analysis 2014*, volume 9017, page 90170L. International Society for Optics and Photonics, 2014.

[102] Fabio Miranda, Lauro Lins, James T Klosowski, and Claudio T Silva. Topkube: A rank-aware data cube for real-time exploration of spatiotemporal data. *IEEE transactions on visualization and computer graphics*, 24(3):1394–1407, 2017.

[103] Mahsa Mirzargar, Ross T Whitaker, and Robert M Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE transactions on visualization and computer graphics*, 20(12):2654–2663, 2014.

[104] Andrew Vande Moere. Time-varying data visualization using information flocking boids. In *IEEE Symposium on Information Visualization*, pages 97–104. IEEE, 2004.

[105] Sophie Morlin-Yron. What's the real size of africa? how western states used maps to downplay size of continent. *CNN. com*, 2019.

[106] Mark A Nearing, Yun Xie, Baoyuan Liu, and Yu Ye. Natural and anthropogenic rates of soil erosion. *International Soil and Water Conservation Research*, 5(2):77–84, 2017.

[107] Gregory Nielson, Hans Hagen, and Heinrich Muller. Scientific visualization. Institute of Electrical & Electronics Engineers, 1997.

[108] Lucy Nowell, Elizabeth Hetzler, and Ted Tanasse. Change blindness in information visualization: A case study. In *Information Visualization, IEEE Symposium on*, pages 15–15. IEEE Computer Society, 2001.

[109] David O'Sullivan, Alastair Morrison, and John Shearer. Using desktop gis for the investigation of accessibility by public transport: an isochrone approach. *International Journal of Geographical Information Science*, 14(1):85–104, 2000.

[110] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[111] Cicero AL Pahins, Sean A Stephens, Carlos Scheidegger, and Joao LD Comba. Hashedcubes: Simple, low memory, real-time visual exploration of big data. *IEEE transactions on visualization and computer graphics*, 23(1):671–680, 2016.

[112] II Pearson. *Map ProjectionsTheory and Applications*. Routledge, 2018.

[113] Donna J Peuquet. It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of american Geographers*, 84(3):441–461, 1994.

[114] Donna J Peuquet and Menno-Jan Kraak. Geobrowsing: creative thinking and knowledge discovery using geographic visualization. *Information Visualization*, 1(1):80–91, 2002.

[115] Kristin Potter, Joe Kniss, Richard Riesenfeld, and Chris R Johnson. Visualizing summary statistics and uncertainty. In *Computer Graphics Forum*, volume 29, pages 823–832. Wiley Online Library, 2010.

[116] Kristin Potter, Andrew Wilson, Peer-Timo Bremer, Dean Williams, Charles Doutriaux, Valerio Pascucci, and Chris Johhson. Visualization of uncertainty and ensemble data: Exploration of climate modeling and weather forecast data with integrated visus-cdat systems. In *Journal of Physics: Conference Series*, volume 180, page 012089. IOP Publishing, 2009.

[117] Kristin Potter, Andrew Wilson, Peer-Timo Bremer, Dean Williams, Charles Doutriaux, Valerio Pascucci, and Chris R Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *2009 IEEE International Conference on Data Mining Workshops*, pages 233–240. IEEE, 2009.

[118] Jordan G Powers, Joseph B Klemp, William C Skamarock, Christopher A Davis, Jimy Dudhia, David O Gill, Janice L Coen, David J Gochis, Ravan Ahmadov, Steven E Peckham, et al. The weather research and forecasting model: Overview, system efforts, and future directions. *Bulletin of the American Meteorological Society*, 98(8):1717–1737, 2017.

[119] Khairi Reda, Chayant Tantipathananandh, Tanya Berger-Wolf, Jason Leigh, and Andrew Johnson. Socioscape–a tool for interactive exploration of spatio-temporal group dynamics in social networks. In *Proc. IEEE Information Visualization Conference (INFOVIS)*, 2009.

[120] Penny Rheingans. Color, change, and control of quantitative data display. In *Proceedings Visualization'92*, pages 252–259. IEEE, 1992.

[121] Salvatore Rinzivillo, Dino Pedreschi, Mirco Nanni, Fosca Giannotti, Natalia Andrienko, and Gennady Andrienko. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3-4):225–239, 2008.

[122] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *IEEE transactions on visualization and computer graphics*, 14(6):1325–1332, 2008.

[123] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260, 1984.

[124] Hanan Samet. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, 2006.

[125] Francesca Samsel, Sebastian Klaassen, Mark Petersen, Terece L Turton, Gregory Abram, David H Rogers, and James Ahrens. Interactive colormapping: Enabling multiple data range and detailed views of ocean salinity. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 700–709. ACM, 2016.

[126] Jibonananda Sanyal, Song Zhang, Jamie Dyer, Andrew Mercer, Philip Amburn, and Robert Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, 2010.

[127] Poonam Shanbhag, Penny Rheingans, et al. Temporal visualization of planning polygons for efficient partitioning of geo-spatial data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 211–218. IEEE, 2005.

[128] Ben Shneiderman. Dynamic queries for visual information seeking. *IEEE software*, 11(6):70–77, 1994.

[129] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pages 364–371. Elsevier, 2003.

[130] Ayush Shrestha, Ben Miller, Ying Zhu, and Yi Zhao. Storygraph: Extracting patterns from spatiotemporal data. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pages 95–103, 2013.

[131] Qingya Shu, Hanqi Guo, Jie Liang, Limei Che, Junfeng Liu, and Xiaoru Yuan. Ensemblegraph: Interactive visual analysis of spatiotemporal behaviors in ensemble simulation data. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 56–63. IEEE, 2016.

[132] Daniel J Simons. Current approaches to change blindness. *Visual cognition*, 7(1-3):1–15, 2000.

[133] WC Skamarock, JB Klemp, J Dudhia, DO Gill, DM Barker, MG Duda, XY Huang, W Wang, and JG Powers. A description of the advanced research wrf version 3, ncar tech note ncar/tn 475 str, 125 pp. *Available from: UCAR Communications, PO Box*, 3000, 2008.

[134] Terry Slocum, Stephen Yoder, Fritz Kessler, and Robert Sluter. Maptime: software for exploring spatiotemporal data associated with point locations. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 37(1):15–32, 2000.

[135] Robert Spencer. Information visualization: Design for interaction. *Person Education*, 2007.

[136] D Stojanovic, S Djordjevic-Kajan, A Mitrovic, and Z Stojanovic. Cartographic visualization and animation of the dynamic geographic processes and phenomena. In *Proceedings of 19th International Cartographic Conference*, volume 1, pages 739–746, 1999.

[137] J Stuart, Joseph Jaquish, Scott Bassett, F Harris, and W Sherman. An interactive visualization method for integrating digital elevation models and geographic information systems vector layers. In *International Symposium on Visual Computing*, pages 553–561. Springer, 2005.

[138] Guodao Sun, Yang Liu, Wenbin Wu, Ronghua Liang, and Huamin Qu. Embedding temporal display into maps for occlusion-free visualization of spatio-temporal data. In *2014 IEEE Pacific Visualization Symposium*, pages 185–192. IEEE, 2014.

[139] Ying Sun and Marc G Genton. Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2):316–334, 2011.

[140] Ying Sun and Marc G Genton. Adjusted functional boxplots for spatio-temporal data visualization and outlier detection. *Environmetrics*, 23(1):54–64, 2012.

[141] Yuzuru Tanahashi and Kwan-Liu Ma. Design considerations for optimizing storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2679–2688, 2012.

[142] Alexandru Telea and Jarke J Van Wijk. Simplified representation of vector fields. In *Proceedings of the conference on Visualization'99: celebrating ten years*, pages 35–42. IEEE Computer Society Press, 1999.

[143] Matthias Thöny, Markus Billeter, and Renato Pajarola. Deferred vector map visualization. In *SIGGRAPH ASIA 2016 Symposium on Visualization*, page 16. ACM, 2016.

[144] Dejan Todorovic. Gestalt principles. *Scholarpedia*, 3(12):5345, 2008.

[145] Christian Tominski, Petra Schulze-Wollgast, and Heidrun Schumann. 3d information visualization for time dependent data on maps. In *Ninth International Conference on Information Visualisation (IV'05)*, pages 175–181. IEEE, 2005.

[146] Edward R Tufte. The visual display of. *Quantitative Information*, page 13, 1983.

[147] Edward R Tufte. *Beautiful evidence*, volume 1. Graphics Press Cheshire, CT, 2006.

[148] Ulanbek D Turdukulov, Menno-Jan Kraak, and Connie A Blok. Designing a visual environment for exploration of time series of remote sensing data: In search for convective clouds. *Computers & Graphics*, 31(3):370–379, 2007.

[149] MJP Vis. History of the mercator projection. B.S. thesis, 2018.

[150] Tatiana Von Landesberger, Sebastian Bremm, Natalia Andrienko, Gennady Andrienko, and Mária Tekušová. Visual analytics methods for categoric spatio-temporal data. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 183–192. IEEE, 2012.

[151] Junpeng Wang, Subhashis Hazarika, Cheng Li, and Han-Wei Shen. Visualization and visual analysis of ensemble data: A survey. *IEEE transactions on visualization and computer graphics*, 25(9):2853–2872, 2018.

[152] Junpeng Wang, Xiaotong Liu, Han-Wei Shen, and Guang Lin. Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE transactions on visualization and computer graphics*, 23(1):81–90, 2016.

[153] Matthew O Ward, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010.

[154] Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing time-series on spirals. In *Infovis*, volume 1, pages 7–14, 2001.

[155] Ross T Whitaker, Mahsa Mirzargar, and Robert M Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2713–2722, 2013.

[156] Susan Wiedenbeck. The use of icons and labels in an end user application program: an empirical study of learning and retention. *Behaviour & Information Technology*, 18(2):68–82, 1999.

[157] Sandra J Winterbottom and David J Gilvear. A gis-based approach to mapping probabilities of river bank erosion: regulated river tummel, scotland. *Regulated Rivers: Research & Management: An International Journal Devoted to River Research and Management*, 16(2):127–140, 2000.

[158] Craig M Wittenbrink, Alex T Pang, and Suresh K Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE transactions on Visualization and Computer Graphics*, 2(3):266–279, 1996.

[159] Keqin Wu and Song Zhang. Visualizing 2d scalar fields with hierarchical topology. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*, pages 141–145. IEEE, 2015.

[160] CC Zhang and K Qin. Theory and methodology of gis spatial analysis, 2004.

[161] Xianfeng Zhang and Micha Pazner. The icon imagemap technique for multivariate geospatial data visualization: approach and software system. *Cartography and Geographic Information Science*, 31(1):29–41, 2004.

[162] Chen Zhong, Tao Wang, Wei Zeng, and Stefan Müller Arisona. Spatiotemporal visualisation: A survey and outlook. In *Digital Urban Modeling and Simulation*, pages 299–317. Springer, 2012.